Robust High Throughput Space-Time Block Coded MIMO Systems

A thesis submitted in fulfilment

of the requirements for the Degree of

Doctor of Philosophy

in Electrical and Computer Engineering

from the University of Canterbury

Christchurch, New Zealand

Nicholas Pau Su Jee

M.E.

June 2007

Abstract

In this thesis, we present a space-time coded system which achieves high throughput and good performance with low processing delay using low-complexity detection and decoding. Initially, Hamming codes are used in a simple interleaved bit-mapped coded modulation structure (BMCM). This is concatenated with Alamouti's orthogonal space-time block codes. The good performance achieved by this system indicates that higher throughput is possible while maintaining performance. An analytical bound for the performance of this system is presented. We also develop a class of low density parity check codes which allows flexible "throughput versus performance" tradeoffs.

We then focus on a Rate 2 quasi-orthogonal space-time block code structure which enables us to achieve an overall throughput of 5.6 bits/symbol period with good performance and relatively simple decoding using iterative parallel interference cancellation. We show that this can be achieved through the use of a bit-mapped coded modulation structure using parallel short low density parity check codes. The absence of interleavers here reduces processing delay significantly. The proposed system is shown to perform well on flat Rayleigh fading channels with a wide range of normalized fade rates, and to be robust to channel estimation errors. A comparison with bit-interleaved coded modulation is also provided (BICM).

Acknowledgments

I would like to express my utmost gratitude to my supervisors, Prof. Des Taylor and Dr. Philippa Martin for their guidance and endless hours of proof-reading. This thesis would not be possible without their continuing support and encouragement over the years. Thank you also to the Foundation for Research, Science and Technology (FRST) for the financial support.

Thank you to the Communications Research Group for their company and Wednesday presentations. Special thanks to Dr. Dave Rankin for his patient advice on coding theory in the early years of my research.

I am greatly indebted to my family, especially my mum, for their endless support over the years. Little things make huge differences. I would like to thank all my friends and people I have met over the years, especially the team at Really Useful Resources, Fiona, Devon and Hayley. Life would be so much more dull without every single one of you.

Nicholas Pau Su Jee

The University of Canterbury

June 2007

Contents

Chapter 1

Introduction

Third-generation (3G) wireless communication systems are required to have high voice quality and provide data bit rates of up to 2 Mbits/s [1]. Recently released fourth-generation (4G) standards will push these rates even higher, possibly over 100Mbits/s [2]. To support such high rates, multiple antennas can be employed to increase channel capacity. A system which employs n_T antennas at the transmitter and n_R antennas at the receiver is called an (n_T, n_R) multiple-input multiple-output (MIMO) system. Information theoretic studies by Telatar [3] and independently by Foschini and Gans [4] showed that MIMO systems can achieve significant channel capacity gains over single-input single-output (SISO) systems ($n_T = 1, n_R = 1$), given that channel state information (CSI) is available at the receiver.

In Section 1.1, a general overview of existing MIMO systems is provided. An important parameter in communication systems is the amount of information that can be transmitted within a defined period of time. This is known as the system throughput, given in bits/symbol period. When bandwidth is also considered, this parameter is known as the spectral efficiency, given in bits/s/Hz. In Section 1.2, we give the definition of these two parameters which are used throughout this thesis. Section 1.3 discusses open problems for MIMO systems and states the aim of this thesis. An outline of the remaining chapters is provided in Section 1.4.

1.1 Overview of MIMO Systems

Wireless channels suffer from severe attenuation due to multipath fading, and interference from other signals. Diversity is the concept of providing less attenuated replica(s) of the transmitted signal. This may occur as temporal diversity using error-correcting codes, or spatial diversity, using multiple antennas. Fig. 1.1 shows a general breakdown of various multiple antenna diversity schemes into categories. The traditional approach is to employ multiple antennas only at the receiver and use maximum ratio combining (MRC). While this improves performance by providing receive diversity, system throughput remains unchanged. In [4], it was shown that capacity grows linearly with $min(n_T, n_R)$. This motivates the use of multiple transmit antennas to increase system throughput.

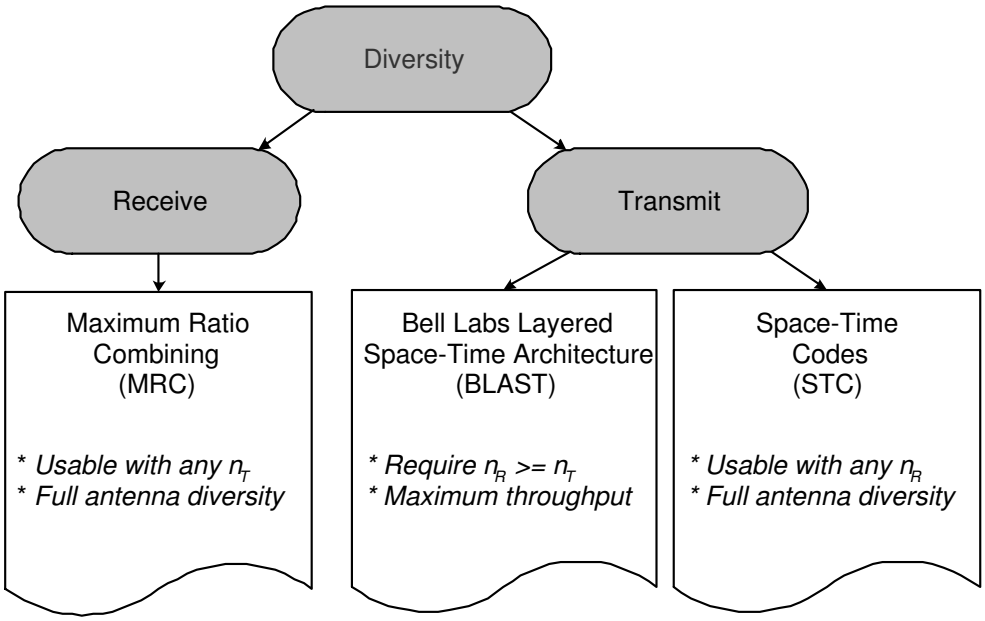


Figure 1.1: Multiple antenna diversity schemes.

In [5], the Bell Labs Layered Space-Time (BLAST) Architecture was proposed to achieve high data rates. BLAST transmits independent data streams simultaneously from each transmit antenna. While BLAST provides maximum throughput[1] for a given number of transmit antennas, it cannot provide full antenna diversity[2]

[1] A scheme is said to provide maximum throughput if it transmits n_T new symbols in each time slot.

[2] A scheme is said to provide full antenna diversity if it achieves a diversity of $n_T n_R$.

for all signals. BLAST also requires the number of receive antennas to be greater than or equal to the number of transmit antennas.

Another approach to using multiple transmit antennas is to employ a space-time code (STC). Although STCs do not provide the maximum throughput possible, they can achieve full antenna diversity and provide better performance compared to BLAST schemes. STCs can be divided into space-time trellis codes (STTCs) and space-time block codes (STBCs). Tarokh et al. first proposed STTCs [6] by jointly designing the channel coding, modulation and transmit diversity. The Viterbi algorithm is applied at the receiver to achieve maximum-likelihood (ML) decoding. STTCs perform very well at the cost of high complexity. When the number of transmit antennas is fixed, decoding complexity increases exponentially with transmission rate [7]. In addressing the issue of decoding complexity, Alamouti discovered a simple STBC [8] for two transmit antennas, which is significantly less complex than STTCs for two transmit antennas. This is an example of orthogonal STBCs (OSTBCs). OSTBCs allow ML decoding to be achieved using simple linear processing. They provide diversity gain, but not coding gain. They also have low rate (except for $n_T = 2$) due to their need to be orthogonal. Higher rates can be achieved using quasi-orthogonal STBCs (QOSTBCs). However, co-channel interference remains with OQSTBCs and this increases detection complexity.

1.2 Throughput and Spectral Efficiency

Traditionally, information is sent using a SISO system. In uncoded SISO systems, M information bits are mapped to one of 2^M constellation symbols. Each symbol is then transmitted using a single transmit antenna within an allocated time slot. The duration of each time slot is matched to the symbol period. We define throughput as the number of information bits that can be transmitted in each time slot (or symbol period). Therefore, the throughput for an uncoded SISO system is M bits/symbol period.

Throughput is a measure of the amount of information we can push through a system within a certain time. It is not a measure of how reliable the information is. In 1948, Claude E. Shannon [9] showed that error-free transmission is achiev-

able if the information is transmitted at a rate below the channel capacity (called the Shannon limit). Forward error-correcting (FEC) codes can be used to improve performance and even approach capacity by adding redundancy to the transmitted signal. This reduces throughput by a factor of $0 \leq R_{fec} \leq 1$, where R_{fec} is the rate of the FEC code. Therefore, the throughput for a coded SISO system is MR_{fec} bits/symbol period.

Throughput can be significantly increased by employing multiple transmit antennas [3, 4]. We can now transmit more[3] than 1 symbol within each time slot. A MIMO system employing a STBC which transmits w new symbols over L symbol periods is said to have rate $R_{stbc} = \frac{w}{L}$, where $R_{stbc} \leq n_T$. STBCs are discussed in more detail in Chapter 2. Each new symbol contains MR_{fec} information bits. The throughput for an FEC coded STBC MIMO system is then

$$Throughput = MR_{stbc}R_{fec} \quad \text{bits/symbol period.} \tag{1.1}$$

If each symbol has period T_s seconds, the symbol rate is $R_s = \frac{1}{T_s}$ symbols/second. The rate of transmitted information bits is then

$$R_{b,info} = R_s MR_{stbc}R_{fec} \quad \text{bits/s.} \tag{1.2}$$

Bandwidth is a very precious resource. Therefore, an important parameter is the information bit rate per channel bandwidth given by

$$\frac{R_{b,info}}{W} \quad \text{bits/s/Hz.} \tag{1.3}$$

Different terminologies for this parameter have been used throughout literature, including spectral (or bandwidth) efficiency [10, 11, 12, 13, 14, 15], throughput [16], capacity [4, 5] and rate [17, 18]. Here, we adopt the term "spectral efficiency" for equation (1.3) as it is a measure of how efficiently we are using the bandwidth to transmit information. Assuming that the symbol rate R_s is equal to the channel bandwidth W, we can write the spectral efficiency of a coded MIMO system as [19]

$$\eta = MR_{stbc}R_{fec} \quad \text{bits/s/Hz.} \tag{1.4}$$

Note that both equations (1.1) and (1.4) contain the same formula but use different units. In this thesis, we adopt the more general throughput definition of equation

[3] Up to a maximum of n_T symbols per time slot.

(1.1), measured in bits/symbol period, as it makes no assumption regarding the symbol rate R_s and channel bandwidth W. System throughput can be easily translated into spectral efficiency by setting $R_s = W$.

1.3 Thesis Contribution

The main goals of employing multiple transmit and receive antennas are to increase throughput and to improve performance over that of single antenna systems. MRC schemes [13, 14] improve performance but provide no throughput gain. Direct transmission of coded modulation schemes [20, 21, 22, 23] in MIMO systems achieves high throughput, at the expense of high detection complexity. For a given number of transmit antennas, maximum throughput can be provided using BLAST schemes [5]. However, these scheme have poor diversity and hence have poor performance. On the other hand, STTCs [6] can be designed to provide maximum diversity and some coding gain. However, their decoding complexity increases exponentially with throughput, making them impractical for high throughput systems. In general, OSTBCs are also designed to provide maximum diversity [8, 24], but they provide no coding gain. They have good performance, but have comparatively low throughput. High rate QOSTBCs have also been proposed in [17, 25], but their decoding complexity increases exponentially with increasing throughput.

The aim of this thesis is to develop MIMO systems that provide:

- high throughput,
- good performance,
- low processing delay,
- detection/decoding schemes with feasible complexity.

We consider different aspects of a MIMO system, including transmit and receive diversity, FEC coding and detection/decoding schemes. An emphasis is placed on low complexity detection and decoding schemes which remain practical with increasing throughput. It should be flexible so that it can be tailored to different system requirements. Robustness to channel estimation errors and time-varying fading is also highly desired.

1.4 Thesis Outline

The rest of the thesis is organised as follows. Chapter 2 provides background information on MIMO systems. It focuses on STBCs (especially high rate QOSTBCs), FEC codes and various MIMO detection schemes.

Chapter 3 develops a very simple interleaved bit-mapped coded modulation (IBMCM) scheme concatenated with the Alamouti OSTBC. Hamming codes are used as component codes in the scheme. The good performance achieved by this simple system indicates that higher throughput is possible, while maintaining good performance at practical decoding complexity. An analytical upper bound on the performance of OSTBCs is also provided.

Chapter 4 describes a new low density parity-check (LDPC) code construction, together with its code properties. LDPC codes are used in this thesis because of their low decoding complexity. They have also been shown to achieve performance very close to channel capacity [26]. The proposed LDPC codes have good girth and are able to provide a wide range of code rates, for a given block length. This provides great flexibility in meeting different system requirements. An in-depth look at the current literature on LDPC codes is also provided in Chapter 4.

The proposed MIMO system, which achieves high throughput and good performance at practical decoding complexity, is described in Chapter 5. LDPC component codes are used in a bit-mapped coded modulation (BMCM) structure which is then concatenated with a QOSTBC. Throughput is increased by using high rate QOSTBCs, which are usually decoded using linear processing and joint detection [17]. Joint detection is optimal, but the decoding complexity increases exponentially with throughput. We propose the use of a suboptimal iterative scheme that has practical decoding complexity, which increases linearly with throughput.

Chapter 6 provides simulation results of the proposed BMCM-QOSTBC system, using the standard quasi-static fading channel assumption and assuming perfect CSI at the receiver. These two assumptions are adopted to allow a fair comparison with existing systems.

In Chapter 7, we investigate the performance of the proposed system in time-varying flat Rayleigh fading, for a wide range of normalized fade rates. In addition,

we demonstrate the robustness of the system to channel estimation errors.

Future work related to the thesis is discussed in Chapter 8, and conclusions are drawn. The original contributions from this PhD research are contained in Chapters 3, 4, 5, 6 and 7. Parts of this work are presented in the following papers:

- N. S. J. Pau, P. A. Martin and D. P. Taylor, *Rate 2 Quasi-Orthogonal Space Time Block Codes Using Parallel Interference Cancellation*, VTC-Spring, Melbourne, Australia, May 2006.

- N. S. J. Pau, D. P. Taylor and P. A. Martin, *Robust High Throughput Space Time Block Codes Using Parallel Interference Cancellation*, submitted to IEEE Transactions on Wireless Communications, June 2006.

Chapter 2

Background

A multiple-input multiple-output (MIMO) system with n_T transmit antennas and n_R receive antennas is said to achieve full diversity if it provides a total diversity order of $n_T n_R$ [6]. Full diversity is desirable because the greater the diversity, the faster the error performance curves fall with increasing signal-to-noise ratio (SNR). A MIMO system achieves maximum transmission rate if it transmits n_T new symbols in each time slot.

The traditional way to combine the received signals from multiple receive antennas is Maximum Ratio Combining (MRC). It is well known that MRC is an optimum combining scheme which provides full receive diversity [27, 14]. The optimum way to transmit multiple signals is debatable. At one end, BLAST schemes provide maximum transmission rate, but suffers from poor transmit diversity. At the other end, space time codes (STCs) can provide full transmit diversity at the cost of transmission rate.

As discussed in Chapter 1, STCs can be divided into two main categories, namely space-time trellis codes (STTCs) and space-time block codes (STBCs). The first STBCs introduced were orthogonal STBCs (OSTBCs) [8], which can be maximum likelihood (ML) decoded using simple linear processing. In this thesis, quasi-orthogonal STBCs (QOSTBCs) are considered. They offer higher rates at the cost of increased detection complexity. Different detection schemes for STBCs are presented in Section 2.2. Section 2.3 compares the performance of various STBCs. In this thesis, system performance is improved by using forward error-correcting (FEC) codes, which are discussed in Section 2.4.

2.1 Space-Time Block Codes

STBCs may be orthogonal or quasi-orthogonal. The symbols in an OSTBC are fully orthogonal to one another, meaning no co-channel interference. In a QOSTBC, some co-channel interference exists but a degree of orthogonality is maintained. In this section, we present the transmission matrices for OSTBCs and QOSTBCs. We use Tx p to denote the p^{th} transmit antenna, where $p = 1, 2, ..., n_T$.

2.1.1 Orthogonal Space-Time Block Codes

A STBC may be represented by a matrix where each column corresponds to a time slot and each row to a transmit antenna. The simplest OSTBC is the Alamouti OSTBC [8] ($n_T = 2$) given by

$$\zeta_{22} = \begin{bmatrix} -s_2^* & s_1 \\ s_1^* & s_2 \end{bmatrix}, \tag{2.1}$$

It takes in two new symbols s_1 and s_2. In time slot 1, s_1 is transmitted from Tx 1 and s_2 from Tx 2. In time slot 2, $-s_2^*$ is transmitted from Tx 1 and s_1^* from Tx 2. The rate of a space-time block code (STBC) is defined as

$$R_{stbc} = \frac{\text{number of new symbols transmitted}, w}{\text{number of time slots used}, L}. \tag{2.2}$$

Therefore, the Alamouti OSTBC has rate 1.

Recall from Section 1.2 that the overall throughput for a system using a STBC and a 2^M-ary constellation is defined as

$$Throughput = MR_{stbc}R_{fec} \qquad \text{bits/symbol period} \tag{2.3}$$

where R_{fec} is the code rate of any FEC codes used. Assuming a symbol rate equivalent to the channel bandwidth, the spectral efficiency of the system is $\eta = MR_{stbc}R_{fec}$ bits/s/Hz.

Tarokh et. al. [24] later generalized the Alamouti OSTBC to $n_T > 2$ transmit antennas using orthogonal designs. It was shown that for real constellations[1], rate 1 OSTBCs exist for arbitrary n_T. However, for complex-constellations, we can only

[1]For example, pulse amplitude modulation (also known as envelope modulation).

guarantee $R_{stbc} = \frac{1}{2}$ OSTBCs for any $n_T > 2$. For the special cases of $n_T = 3$ and $n_T = 4$, complex-constellation OSTBCs exist for $R_{stbc} = \frac{3}{4}$ [24, 18].

The Rate-1/2 Tarokh OSTBC [24] for $n_T = 4$ is given by

$$\zeta_{48} = \begin{bmatrix} s_4^* & -s_3^* & -s_2^* & s_1^* & s_4 & -s_3 & -s_2 & s_1 \\ -s_3^* & -s_4^* & s_1^* & s_2^* & -s_3 & -s_4 & s_1 & s_2 \\ -s_2^* & s_1^* & -s_4^* & s_3^* & -s_2 & s_1 & -s_4 & s_3 \\ s_1^* & s_2^* & s_3^* & s_4^* & s_1 & s_2 & s_3 & s_4 \end{bmatrix}, \tag{2.4}$$

It transmits $w = 4$ new symbols in $L = 8$ time slots using 4 antennas, giving $R_{stbc} = \frac{1}{2}$. This results in a spectral efficiency loss of 50 percent compared to the Alamouti OSTBC.

Foschini and Gans [4] showed that MIMO systems can achieve significantly higher throughput than SISO systems. Spatial multiplexing MIMO systems, such as BLAST, achieve high throughput but generate high levels of co-channel interference, which degrades performance. OSTBCs are popular because they achieve full diversity and good performance for low detection complexity. This is because the orthogonality in the OSTBC allows the receiver to eliminate the co-channel interference using only linear processing. However, OSTBCs have $R_{stbc} < 1$ for $n_T > 2$. This reduces system throughput, which contradicts the original goal of using MIMO systems.

2.1.2 Quasi-Orthogonal Space-Time Block Codes

STBCs with $R_{stbc} \geq 1$ for $n_T > 2$ are highly desirable because the system throughput increases with R_{stbc}. This increase may be achieved by sacrificing some orthogonality, resulting in QOSTBCs. The QOSTBC in [17] is constructed using the Alamouti OSTBCs from equation (2.1) and is given by [25]

$$\zeta_{44} = \begin{bmatrix} -\zeta_{22}^*(s_3, s_4) & \zeta_{22}(s_1, s_2) \\ \zeta_{22}^*(s_1, s_2) & \zeta_{22}(s_3, s_4) \end{bmatrix} = \begin{bmatrix} s_4 & -s_3^* & -s_2^* & s_1 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ s_1 & s_2^* & s_3^* & s_4 \end{bmatrix}, \tag{2.5}$$

where $\zeta_{22}(s_1, s_2)$ and $\zeta_{22}(s_3, s_4)$ are Alamouti OSTBCs for the symbol pairs (s_1, s_2) and (s_3, s_4), respectively. This is also known as the extended Alamouti scheme [25].

It transmits four new symbols (s_1, s_2, s_3, s_4) in each block of $L = 4$ time slots using $n_T = 4$ transmit antennas. Therefore, it has $R_{stbc} = 1$ and we refer to it as the Rate 1 QOSTBC.

In order to achieve R_{stbc} greater than 1, we consider the double space-time transmit-diversity (DSTTD) scheme of [16] which can be described by

$$\zeta_{42} = \begin{bmatrix} -s_2^* & s_1 \\ s_1^* & s_2 \\ -s_4^* & s_3 \\ s_3^* & s_4 \end{bmatrix}, \tag{2.6}$$

where each group of $w = 4$ new constellation points (s_1, s_2, s_3, s_4) is transmitted using $L = 2$ time slots and four transmit antennas. We note that the DSTTD scheme is a truncated version of the Rate 1 QOSTBC in equation (2.5). If we split the transmit antennas into Group A (Tx1, Tx2) and Group B (Tx3, Tx4), we are effectively transmitting the two groups independently (as in BLAST schemes), where each group uses the simple Alamouti OSTBC. Therefore, DSTTD is quasi-orthogonal and has $R_{stbc} = 2$, so we shall refer to it as the Rate 2 QOSTBC. Note that the quasi-orthogonal schemes generate co-channel interference that cannot be mitigated using linear processing.

2.2 Detection Schemes

The symbol sent from transmit antenna p in time slot t, $x_p(t)$, to receive antenna q experiences multipath fading, denoted $\alpha_p^q(t)$. The quasi-static fading assumption means that $\alpha_p^q(1) = \alpha_p^q(2) = ...\alpha_p^q(L)$ within each STBC block of length L. The faded symbols from all n_T transmit antennas are superimposed to form the received signal at the q^{th} receive antenna as

$$r_q(t) = \sum_{p=1}^{n_T} \alpha_p^q(t) x_p(t) + w_q(t), \qquad t = 1, ..., L \tag{2.7}$$

where $w_q(t)$ is the additive white Gaussian noise (AWGN) at the q^{th} receive antenna.

STBCs are initially decoded using linear processing. This simple detection process is described in Section 2.2.1. For OSTBCs, this achieves maximum-likelihood (ML) decoding as each symbol is completely decoupled from every other symbol. For

QOSTBCs, co-channel interference remains following linear processing. In [17], a pairwise joint detection (JD) scheme is used to minimize the effect of this interference. Successive interference cancellation (SIC) schemes, which are widely used in BLAST systems, can also be used for QOSTBCs [16]. Here, we propose the use of parallel interference cancellation (PIC) schemes to minimize the effects of the co-channel interference. PIC schemes are widely used in multi-user environments [28] and have smaller processing delays than SIC schemes [29, 30].

2.2.1 Linear Processing

At the receiver, linear processing [8, 24] is first applied to the received signals to produce an estimate of the transmitted symbols. We will initially use the Alamouti OSTBC as an example. The received signals in time slots 1 and 2 at receive antenna q are given by

$$\begin{aligned} r_q(1) =& \alpha_1^q(1)s_1 + \alpha_2^q(1)s_2 + w_q(1) \\ r_q(2) =& -\alpha_1^q(2)s_2^* + \alpha_2^q(2)s_1^* + w_q(2) \end{aligned} \tag{2.8}$$

where $w_q(t)$ is the receiver noise at receive antenna q in time slot t. Assuming perfect CSI is available at the receiver, linear processing produces estimates of the transmitted symbols s_1, s_2 given by

$$\begin{aligned} \hat{s}_1 =& \sum_{q=1}^{n_R} (\alpha_1^{q*} r_q(1) + \alpha_2^q r_q^*(2)) \\ \hat{s}_2 =& \sum_{q=1}^{n_R} (\alpha_2^{q*} r_q(1) - \alpha_1^q r_q^*(2)). \end{aligned} \tag{2.9}$$

The summation in equation (2.9) results from the use of MRC to combine the received signals from all n_R receive antennas. The time notation is dropped from the α_p^q terms because $\alpha_p^q(1) = \alpha_p^q(2)$ due to the quasi-static fading assumption which requires the channel to be constant for $L = 2$ time slots. Substituting equation (2.8) into equation (2.9) gives

$$\begin{aligned} \hat{s}_1 =& \sum_{q=1}^{n_R} (|\alpha_1^q|^2 + |\alpha_2^q|^2) s_1 + \sum_{q=1}^{n_R} (\alpha_1^{q*} w_q(1) + \alpha_2^q w_q^*(2)) \\ \hat{s}_2 =& \underbrace{\sum_{q=1}^{n_R} (|\alpha_1^q|^2 + |\alpha_2^q|^2) s_2}_{\text{desired terms}} + \underbrace{\sum_{q=1}^{n_R} (\alpha_2^{q*} w_q(1) - \alpha_1^q w_q^*(2))}_{\text{noise terms}}, \end{aligned} \tag{2.10}$$

where the estimates are simply the transmitted symbols scaled by the sum of squared fading coefficients and corrupted by some noise terms. Let χ be the set of 2^M constellation symbols. The decision rule is to choose $s_i = \beta$ to minimize

$$||\hat{s}_i - K\beta||^2 \tag{2.11}$$

where $\beta \in \chi$ and K is the coefficient of s_i after linear processing. For equal energy constellations such as M-PSK, this decision rule is ML. Since the orthogonality of the Alamouti OSTBC allows the symbols to be decoupled at the receiver, the minimization process can be performed independently for s_1 and s_2 without losing performance. This reduces the detection complexity from $(2^M)^2$ to $2(2^M)$ searches.

For $n_T = 4$, linear processing produces estimates of the transmitted symbols s_1, s_2, s_3, s_4, given by

$$\begin{aligned}
\hat{s}_1 =& 2\sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_1 + noise \\
\hat{s}_2 =& 2\sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_2 + noise \\
\hat{s}_3 =& 2\sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_3 + noise \\
\hat{s}_4 =& \underbrace{2\sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_4}_{\text{desired terms}} + noise.
\end{aligned} \tag{2.12}$$

for the Rate 1/2 OSTBC,

$$\begin{aligned}
\hat{s}_1 =& \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_1 + \sum_{q=1}^{n_R} 2(\alpha_1^q\alpha_4^{q*} - \alpha_2^q\alpha_3^{q*})s_4 + noise \\
\hat{s}_2 =& \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_2 + \sum_{q=1}^{n_R} 2(\alpha_2^q\alpha_3^{q*} - \alpha_1^q\alpha_4^{q*})s_3 + noise \\
\hat{s}_3 =& \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_3 + \sum_{q=1}^{n_R} 2(\alpha_2^q\alpha_3^{q*} - \alpha_1^q\alpha_4^{q*})s_2 + noise \\
\hat{s}_4 =& \underbrace{\sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_4}_{\text{desired terms}} + \underbrace{\sum_{q=1}^{n_R} 2(\alpha_1^q\alpha_4^{q*} - \alpha_2^q\alpha_3^{q*})s_1}_{\text{interference terms}} + noise.
\end{aligned} \tag{2.13}$$

for the Rate 1 QOSTBC and

$$\begin{aligned}
\hat{s}_1 &= \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2)s_1 + \sum_{q=1}^{n_R}(\alpha_1^{q*}\alpha_3^q + \alpha_2^q\alpha_4^{q*})s_3 + \sum_{q=1}^{n_R}(\alpha_1^{q*}\alpha_4^q - \alpha_2^q\alpha_3^{q*})s_4 + noise \\
\hat{s}_2 &= \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2)s_2 + \sum_{q=1}^{n_R}(\alpha_2^{q*}\alpha_4^q + \alpha_1^q\alpha_3^{q*})s_4 + \sum_{q=1}^{n_R}(\alpha_2^{q*}\alpha_3^q - \alpha_1^q\alpha_4^{q*})s_3 + noise \\
\hat{s}_3 &= \sum_{q=1}^{n_R}(|\alpha_3^q|^2 + |\alpha_4^q|^2)s_3 + \sum_{q=1}^{n_R}(\alpha_3^{q*}\alpha_1^q + \alpha_4^q\alpha_2^{q*})s_1 + \sum_{q=1}^{n_R}(\alpha_3^{q*}\alpha_2^q - \alpha_4^q\alpha_1^{q*})s_2 + noise \\
\hat{s}_4 &= \underbrace{\sum_{q=1}^{n_R}(|\alpha_3^q|^2 + |\alpha_4^q|^2)s_4}_{\text{desired terms}} + \underbrace{\sum_{q=1}^{n_R}(\alpha_4^{q*}\alpha_2^q + \alpha_3^q\alpha_1^{q*})s_2 + \sum_{q=1}^{n_R}(\alpha_4^{q*}\alpha_1^q - \alpha_3^q\alpha_2^{q*})s_1}_{\text{interference terms}} + noise.
\end{aligned} \tag{2.14}$$

for the Rate 2 QOSTBC. These equations assume quasi-static fading so the time notation is again dropped from the α_p^q terms. For simplicity, we group all the noise terms together.

Since the Rate 1/2 OSTBC of equation (2.4) is orthogonal, the linear processing estimates in equation (2.12) are simply a scaled version of the decoupled transmitted symbols plus some noise terms. Therefore, the minimization process in the decision rule in equation (2.11) can be performed individually for s_1, s_2, s_3 and s_4. This requires a search through $4(2^M)$ constellation symbols in total.

The presence of co-channel interference terms in equations (2.13) and (2.14) is due to the lack of full orthogonality in the QOSTBC structures of equations (2.5) and (2.6). If the decision rule of equation (2.11) is applied immediately after linear processing, performance will thus be severely degraded. This is because the interference terms would be treated as noise, but they remain the same even with increasing SNR. Therefore, it is necessary to compensate the effects of these interference terms.

For QOSTBCs, the symbol estimates from the linear processing stage are used by either the joint detection (JD) scheme of [17] or the succesive interference cancellation (SIC) scheme of [16] to produce better estimates by reducing the co-channel interference, as explained in Sections 2.2.2 and 2.2.3, respectively. In this thesis, we propose the use of parallel interference cancellation (PIC) as an alternative way to reduce co-channel interference. The PIC scheme is explained in Section 2.2.4. Here all three schemes are described using the Rate 1 QOSTBC. The extension to the Rate 2 QOSTBC is straightforward.

2.2.2 Joint Detection

In [17], joint detection (JD) is used to detect the transmitted symbols for the Rate 1 QOSTBC. The symbol estimates after linear processing are given in equation (2.13). Note that the symbols within the pairs (s_1, s_4) and (s_2, s_3) interfere with each other, but symbols between the pairs do not. Therefore, pairwise JD can be performed on the pair (s_1, s_4) independently from the pair (s_2, s_3). For symbols s_1 and s_4, the decision rule using pairwise JD is to choose $s_1 = \beta$ and $s_4 = \gamma$ to jointly minimize the metric

$$||\hat{s}_1 - K_1\beta - K_{1,int}\gamma||^2 + ||\hat{s}_4 - K_4\gamma - K_{4,int}\beta||^2 \tag{2.15}$$

where K_1 and $K_{1,int}$ are the coefficients of the desired term and interference term, respectively, for s_1 and K_4 and $K_{4,int}$ are the coefficients of the desired term and interference term, respectively, for s_4.

Similarly for symbols s_2, s_3, the decision rule is to choose $s_2 = \beta$ and $s_3 = \gamma$ to jointly minimize the metric

$$||\hat{s}_2 - K_2\beta - K_{2,int}\gamma||^2 + ||\hat{s}_3 - K_3\gamma - K_{3,int}\beta||^2 \tag{2.16}$$

where K_2 and $K_{2,int}$ are the coefficients of the desired term and interference term, respectively, for s_2 and K_3 and $K_{3,int}$ are the coefficients of the desired term and interference term, respectively, for s_3. This requires a search through $2(2^M)^2$ pairs (β, γ) of constellation symbols.

For the Rate 2 QOSTBC, all four symbols interfere with each other to a certain degree[2]. Therefore, JD must be performed for all four symbols, requiring a search through $(2^M)^4$ combinations of constellation symbols in total.

2.2.3 Successive Interference Cancellation

SIC is widely used in multiuser environments [28, 29]. SIC works by successively detecting the k^{th} user, and then subtracting the interference due to that user from the signals for the other users. When the $(k+1)^{th}$ user is detected, the remaining signal contains one less interference term than the k^{th} user. This process is repeated until the last user is detected.

[2]For example, from equation (2.14), s_1 contains interference terms from s_3 and s_4. However, s_3 and s_4 both contain interference from s_2. Therefore, s_1 contains interference from s_2, s_3 and s_4

SIC was adopted to avoid the complexity of ML detection for the BLAST scheme proposed in [5] where co-channel interference terms were treated as interfering users. SIC can also be applied to QOSTBCs, where it is used to minimize the effect of the interference terms in equations (2.13) and (2.14).

For the Rate 1 QOSTBC, each symbol estimate contains one interference term. We know that the symbols s_1, s_4 are independent of s_2, s_3. First we describe SIC on s_1 and s_4. SIC is performed in stages. Let the soft estimates from linear processing be $\hat{s}_1^0$ and $\hat{s}_4^0$ where this initial stage prior to SIC is denoted stage $\kappa = 0$. We detect these two estimates independently using the decision rule in equation (2.11), thereby treating the interference terms as noise. This requires a search through 2^M constellation symbols for each estimate. The results of the search are two hard decisions $\hat{s}_{1,h}^0$ and $\hat{s}_{4,h}^0$.

In stage 1 of SIC, we cancel the interference term s_4 from $\hat{s}_1^0$ according to

$$\hat{s}_1^1 = \hat{s}_1^0 - K_{1,int}\hat{s}_{4,h}^0 \tag{2.17}$$

where the hard estimate $\hat{s}_{4,h}^0$ from stage 0 is used as an estimate of the interference s_4. $K_{i,int}$ is the interference term coefficient of $\hat{s}_i$ in equation (2.13). Then we detect $\hat{s}_{1,h}^1$ using equation (2.11). Next, we cancel out the interference term s_1 from $\hat{s}_4^0$ according to

$$\hat{s}_4^1 = \hat{s}_4^0 - K_{4,int}\hat{s}_{1,h}^1 \tag{2.18}$$

where the hard estimate $\hat{s}_{1,h}^1$ is used to represent the interference s_1. Note that $\hat{s}_{1,h}^1$ is used as it is more up to date than $\hat{s}_{1,h}^0$. Finally in stage 1, we detect $\hat{s}_{4,h}^1$ using equation (2.11). Therefore, stage 1 consists of four steps where the steps must be performed successively. Subsequent stages[3] are simply a repeat of Stage 1, except the stage index κ is incremented by 1 until a predetermined number of SIC stages is reached, or until the decisions converge.

As discussed, the detection and cancellation processes for s_2 and s_3 can be independently performed in parallel to that for s_1 and s_4. Therefore, each SIC stage requires 4 time steps and 2 processors for all four symbols. This is a special case and is limited to the Rate 1 QOSTBC. For the Rate 2 QOSTBC, each symbol contains interference from every other transmitted symbol to a certain degree. It is trivial

[3]Although SIC is generally carried out using only one stage [4, 30], it has been shown that additional stages improve performance [28, 31, 32].

to show that each SIC stage for the Rate 2 QOSTBC requires 8 time steps and 1 processor for all four symbols. The detection complexity in each SIC stage is given by $4(2^M)$ searches, for both Rate 1 and Rate 2 QOSTBCs.

The choice of s_1 over s_4 as the first symbol to be detected in each stage for the Rate 1 QOSTBC is arbitrary since the desired symbol in the estimates in equation (2.13) has the same coefficient K_i and hence the same power, for each s_i. In the Rate 2 QOSTBC, however, the coefficients of the desired symbols, K_i's, in equation (2.14) are not equal. Therefore, ordering is required and the symbol corresponding to the largest K_i should be detected first, followed by the next largest and so on.

2.2.4 Parallel Interference Cancellation

PIC is widely used in multiuser environments. Although it requires more processors to perform tasks in parallel, considerable savings in processing delays are achieved [29, 30]. For the Rate 1 QOSTBC, each symbol estimate contains one interference term. Like SIC, PIC is also performed in stages. Let the soft estimates from linear processing be $\hat{s}_i^{\kappa=0}$ where this initial stage before PIC is denoted by $\kappa = 0$. We detect each of these estimates independently using the decision rule in equation (2.11), thereby treating the interference terms as noise. This requires a search through 2^M constellation symbols for each estimate. The result of the search are hard decisions, $\hat{s}_{i,h}^0$.

In stage 1 of PIC, we cancel out all the interference terms in equation (2.13) according to

$$\begin{aligned}
\hat{s}_1^1 &= \hat{s}_1^0 - K_{1,int}\hat{s}_{4,h}^0 \\
\hat{s}_2^1 &= \hat{s}_2^0 - K_{2,int}\hat{s}_{3,h}^0 \\
\hat{s}_3^1 &= \hat{s}_3^0 - K_{3,int}\hat{s}_{2,h}^0 \\
\hat{s}_4^1 &= \hat{s}_4^0 - K_{4,int}\hat{s}_{1,h}^0
\end{aligned} \tag{2.19}$$

where the hard decision $\hat{s}_{i,h}^0$ from stage 0 is used to represent the interference of s_i. Then we detect each $\hat{s}_{i,h}^1$ using equation (2.11). Subsequent stages are simply a repeat of Stage 1, except the stage index κ is incremented by 1 until a predetermined number of PIC stages is reached or until the performance converges.

For the Rate 1 QOSTBC, each stage consists of only 2 time steps but 4 proces-

sors are required, one for each symbol. It is trivial to show that the Rate 2 QOSTBC has the same time step and processor requirements. No ordering is required for any PIC scheme since interference cancellation is performed simultaneously.

2.2.5 Detection Complexity Comparison

The Rate 1 QOSTBC under consideration transmits $w = 4$ new symbols in $L = 4$ time slots. In terms of detection complexity, we have shown that JD benefits from the pairwise decoupling of these 4 symbols after linear processing. Pairwise JD requires $2(2^M)^2$ searches compared to only $4\kappa(2^M)$ for both the SIC and PIC schemes. Interference cancellation schemes do not benefit from pairwise decoupling as symbol detection is performed individually in each stage.

The Rate 1 QOSTBC is a special case because of its pairwise decoupling of symbol estimates. This vanishes as we increase R_{stbc} beyond 1 due to the increasing number of interference terms. For QOSTBCs transmitting w new symbols in L time slots at $R_{stbc} > 1$, JD requires a search through $(2^M)^w$ combinations of constellation symbols. For both the SIC and PIC schemes, symbol detection is performed individually in each stage. Therefore, a total of $w\kappa(2^M)$ searches is required for either scheme, where κ is the number of stages used. For the Rate 2 QOSTBC using 16-QAM, JD requires 65536 searches in total, compared to only 64κ searches for interference cancellation schemes. For 64-QAM, these numbers increase to 16,777,216 for JD compared to only 256κ for interference cancellation schemes. Typically, setting $4 \leq \kappa \leq 9$ provides good performance, as will be shown in Chapter 6.

Fig. 2.1 shows the complexity of JD compared to the interference cancellation schemes, when κ is set to 5. Complexity is measured as the total number of searches required to detect all w QOSTBC symbols for a constellation of size 2^M. Note the exponential scale on the vertical axis. The search complexity for both JD and the interference cancellation schemes increases linearly with constellation size. JD has lower complexity for BPSK constellations as the interference cancellation schemes use $\kappa = 5$, compared to only 1 for JD. The complexity of JD increases faster with constellation size, compared to interference cancellation schemes. Since we are interested in high throughput schemes where $M \geq 2$, interference cancellation schemes, although suboptimal, are preferred to JD.

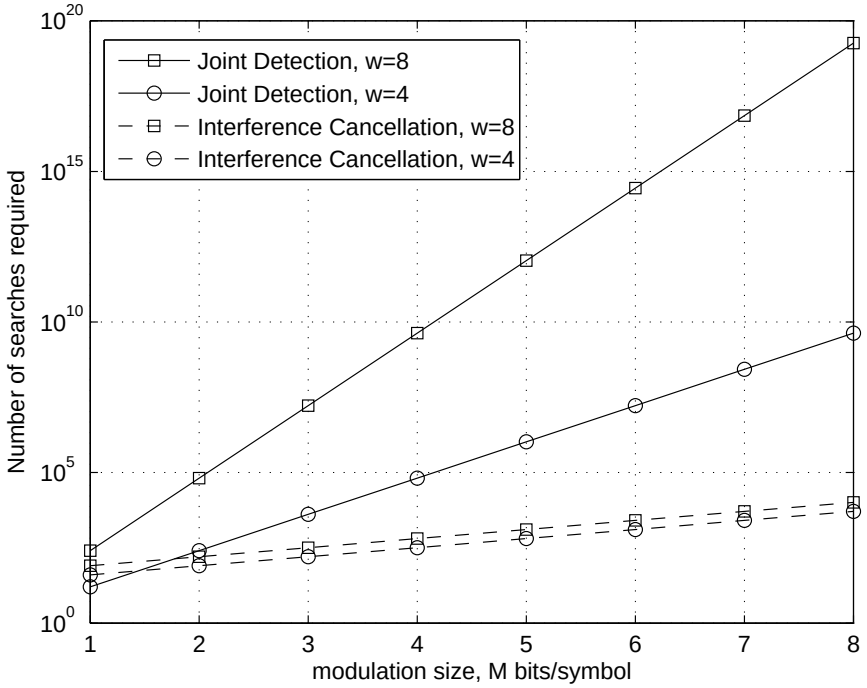


Figure 2.1: Complexity comparison between joint detection and interference cancellation schemes.

For $n_T = 2$, we use the Alamouti OSTBC and have $w = 2$. When $n_T = 4$, we use the Tarokh Rate 1/2 OSTBC, the Rate 1 QOSTBC, and the Rate 2 QOSTBC, and have $w = 4$. Typically, w increases with n_T. Fig. 2.1 shows that JD complexity grows exponentially with w, compared to a linear growth for interference cancellation schemes. Therefore, as MIMO research moves to systems with a growing number of transmit antennas, interference cancellation schemes becomes an increasingly attractive alternative to JD.

2.3 STBC Performance

Fig. 2.2 shows the BER performance using Monte Carlo simulation of the Alamouti OSTBC with $n_T = 2$ and $n_R = 1, 2$; and MRC schemes with $n_T = 1$ and $n_R = 1, 2, 4$. QPSK modulation is used in all cases, giving a throughput of 2 bits/symbol period. At a diversity order of 2, the Alamouti OSTBC with $n_T = 2$ and $n_R = 1$ has the same slope as that of an MRC scheme with $n_T = 1$ and $n_R = 2$. Similarly at a

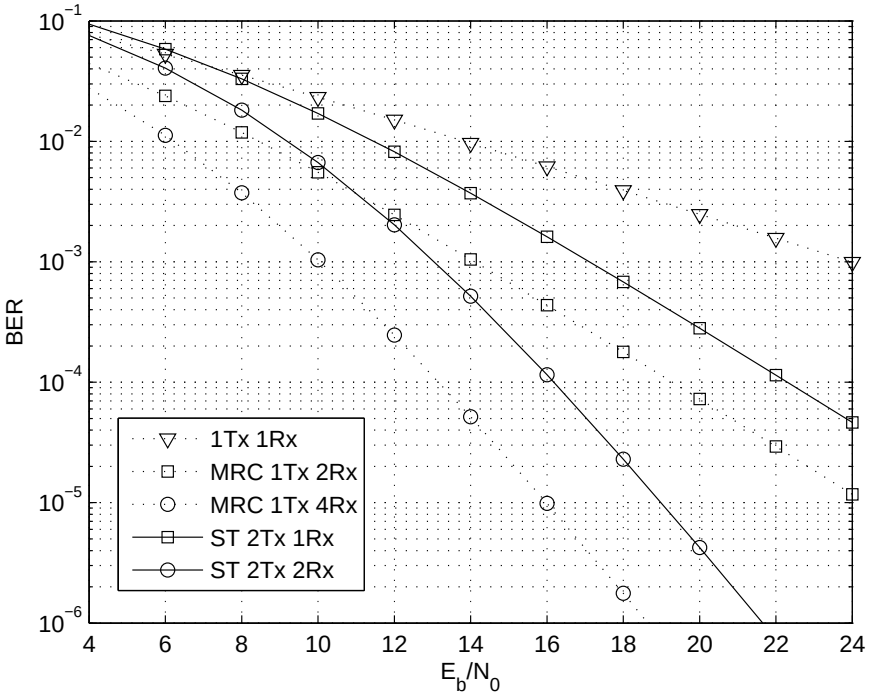


Figure 2.2: BER performance comparison between Alamouti OSTBC schemes and MRC schemes for QPSK. Throughput is 2 bits/symbol period.

diversity order of 4, the Alamouti OSTBC with $n_T = 2$ and $n_R = 2$ has the same slope as the MRC scheme with $n_T = 1$ and $n_R = 4$. However, the Alamouti OSTBC suffers from a signal-to-noise ratio (SNR) loss.

Space-time codes are usually designed to achieve full diversity [24, 6]. No coding gain is provided by STBCs [24]. The diversity of the STBC determines the slope of the performance curves whereas coding gain shifts them horizontally to the left. Since MRC achieves full diversity, the Alamouti OSTBC also achieves full diversity, as evidenced by the same slopes at the same diversity orders. However, the Alamouti OSTBC incurs a 3dB power penalty, compared to MRC with an equivalent diversity order [24]. This is because the total transmit power when using the OSTBC is normalized to 1, resulting in each antenna transmitting with half the power of the single transmit antenna in MRC schemes.

Fig. 2.3 shows the BER performance comparison for OSTBC systems with diversity order 4. 16-QAM modulation is used in conjunction with the Rate 1/2 Tarokh OSTBC, Alamouti OSTBC and MRC schemes, giving throughputs of 2, 4

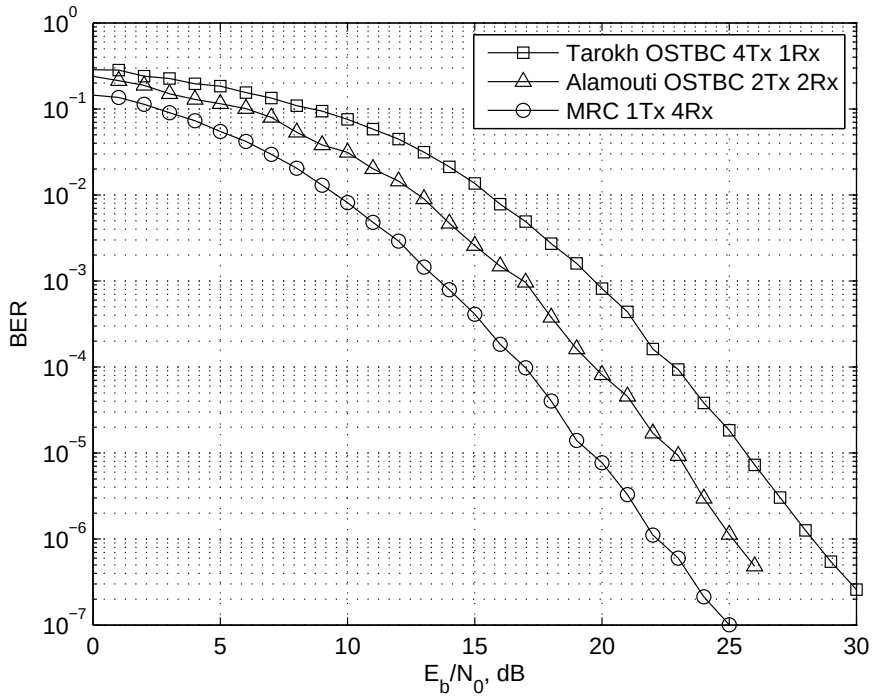


Figure 2.3: BER performance using 16-QAM for the MRC scheme with $n_T = 1$ and $n_R = 4$, the Alamouti OSTBC with $n_T = 2$ and $n_R = 2$ and the Rate 1/2 OSTBC with $n_T = 4$ and $n_R = 1$. All systems have a diversity order of 4.

and 4 bits/symbol period, respectively. Direct transmission schemes are essentially Rate 1 OSBTCs with a single element $[s_1]$. Again, we observe that the Alamouti OSTBC incurs a 3dB power penalty, compared to the MRC scheme with the same diversity order. The Rate 1/2 Tarokh OSTBC, however, incurs a 6dB power penalty compared to the MRC scheme. This is because each transmit antenna in the Rate 1/2 Tarokh OSTBC transmits with only a quarter of the power compared to direct transmission in the MRC scheme. We can show that Rate 1/2 OSTBCs with a single receive antenna suffer a $3log_2(n_T)$dB power penalty compared to MRC schemes with a single transmit antenna of equivalent diversity orders i.e. when $n_T = n_R$. The OSTBCs have the same diversity as the MRC scheme, verifying the claim in [24] that OSTBCs achieve full diversity.

Fig. 2.4 shows the BER performance comparison between the Rate 1/2 OSTBC using 16-QAM and the Rate 1 QOSTBC using QPSK. Throughput is 2 bits/symbol period. Linear processing of the OSTBC decouples the transmitted symbols, allowing each symbol to be detected individually. The symbol estimates

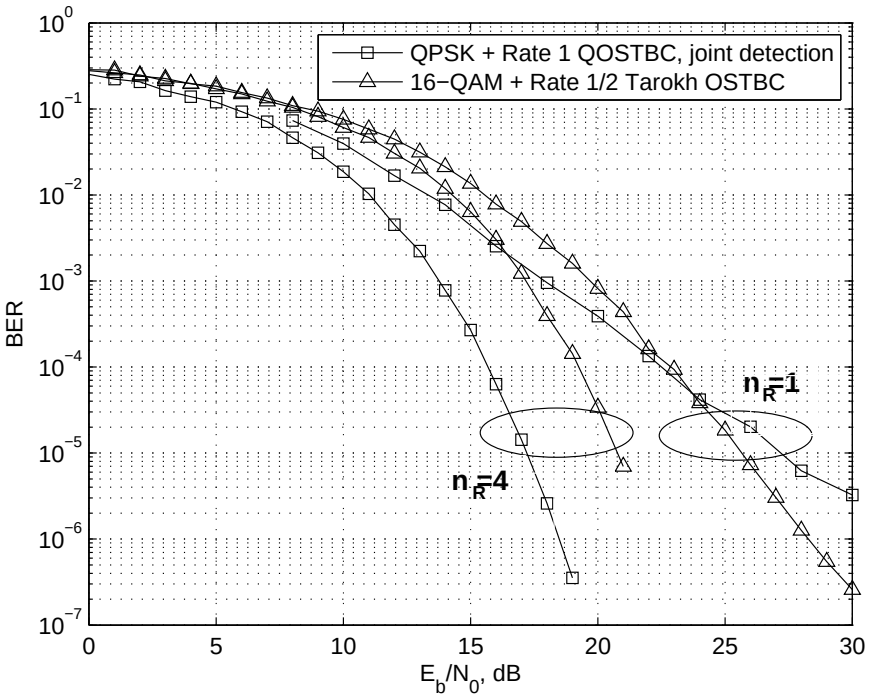


Figure 2.4: BER performance comparison for the Rate 1/2 Tarokh OSTBC using 16-QAM and the Rate 1 QOSTBC using QPSK, for $n_T = 4$ and $n_R = 1, 4$. Throughput is 2 bits/symbol period.

for OSTBCs are free from interference. However, the symbol estimates for QOSTBCs contain one interference term each. Hence for the QOSTBC, linear processing is followed by pairwise JD to minimize the effect of this interference.

When $n_T = 4$ and $n_R = 1$, the QOSTBC outperforms the OSTBC at low SNR. This is because, for the same throughput, the higher rate of the QOSTBC allows the use of a smaller modulation size (QPSK) which has better minimum Euclidean distance to protect against noise at low SNR. At high SNR, the OSTBC outperforms the QOSTBC with a crossover at around BER=4×10^{-5}. The interference free OSTBC achieves full diversity whereas the QOSTBC only achieves half the full diversity. The greater the diversity, the steeper the performance curves fall at high SNR.

When $n_T = 4$ and $n_R = 4$, the QOSTBC outperforms the OSTBC down to at least BER=10^{-7}. Further increasing the diversity provides a diminishing increase in effective SNR gain (c.f. Appendix A). Since the diversity order of the system is now increased to 16, the additional diversity of the OSTBC becomes insignificant

and does not provide much SNR gain. On the other hand, the additional protection against noise provided by the smaller constellation used with the QOSTBC provides a 3.8dB gain over the OSTBC.

2.4 Forward Error Correcting Codes

In Section 2.1.2, we showed how system throughput can be increased by increasing R_{stbc}. However, system performance is degraded when $R_{stbc} > 1$ due to increasing levels of co-channel interference. Three different approaches to minimizing the effect of this interference were described in Section 2.2. Here, we describe yet another way to improve system performance, namely the use of forward error correcting (FEC) codes. In this section, we summarize some basic properties of block codes. In Chapter 4, we develop high performance block codes, called low-density parity check (LDPC) codes.

2.4.1 Generator and Parity-Check Matrices

A (n,k) binary FEC code encodes a block of k information bits into a block of n codeword bits. In this thesis, only linear binary codes are considered. A code is linear if and only if the sum of any two codewords forms another codeword [33]. The sum of two binary codewords is their vector addition modulo two. Since a linear code forms a vector subspace, it can be described by a basis set of k linearly independent codewords. This basis set forms a k-by-n generator matrix $\boldsymbol{G}$. As a subspace may have more than one basis, so a code may have more than one generator matrix. Using row reduction methods [33], any generator matrix can be written in the systematic form

$$\boldsymbol{G} = [\boldsymbol{I}_k | \boldsymbol{P}]. \tag{2.20}$$

where $\boldsymbol{I}_k$ is a k-by-k identity matrix and $\boldsymbol{P}$ is a k-by-$(n-k)$ matrix. The encoding process is given by $\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G}$ where $\boldsymbol{u}$ is a 1-by-k information vector, $\boldsymbol{c}$ is a 1-by-n codeword vector.

For example, the systematic generator matrix for a simple (7,4) Hamming

code is

$$\boldsymbol{G_{ham74}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{2.21}$$

This generates a set of $2^4 = 16$ codewords given by

$$\begin{array}{ll} \boldsymbol{u} \rightarrow \boldsymbol{c} & \boldsymbol{u} \rightarrow \boldsymbol{c} \\ 0000 \rightarrow 0000000 & 1000 \rightarrow 1000101 \\ 0001 \rightarrow 0001110 & 1001 \rightarrow 1001011 \\ 0010 \rightarrow 0010011 & 1010 \rightarrow 1010110 \\ 0011 \rightarrow 0011101 & 1011 \rightarrow 1011000 \\ 0100 \rightarrow 0100111 & 1100 \rightarrow 1100010 \\ 0101 \rightarrow 0101001 & 1101 \rightarrow 1101100 \\ 0110 \rightarrow 0110100 & 1110 \rightarrow 1110001 \\ 0111 \rightarrow 0111010 & 1111 \rightarrow 1111111 \end{array} \tag{2.22}$$

where the information vector $\boldsymbol{u}$ is mapped to codeword vectors $\boldsymbol{c}$. Let $(c_1, c_2, c_3, c_4, c_5, c_6, c_7)$ be any codeword in (2.22). Then the parity bits (c_5, c_6, c_7) can be computed using modulo-2 addition of the information bits (c_1, c_2, c_3, c_4), as given by

$$\begin{aligned} c_5 &= c_1 + c_2 + c_4 \\ c_6 &= c_2 + c_3 + c_4 \\ c_7 &= c_1 + c_2 + c_3. \end{aligned} \tag{2.23}$$

These are the parity check equations for the Hamming code. They can be expressed in terms of the parity check matrix $\boldsymbol{H}$

$$\boldsymbol{H_{ham74}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.24}$$

The parity check matrix $\boldsymbol{H}$ is another way to define a code and in systematic form is given by

$$\boldsymbol{H} = [\boldsymbol{P}^T | \boldsymbol{I}_{n-k}]. \tag{2.25}$$

where $\boldsymbol{P}^T$ is the transpose of the submatrix $\boldsymbol{P}$ from the systematic generator matrix $\boldsymbol{G}$ in equation (2.20). Every codeword is orthogonal to the rows of $\boldsymbol{H}$. Therefore,

$$\boldsymbol{G}\boldsymbol{H}^T = 0 \quad \text{and} \quad \boldsymbol{c}\boldsymbol{H}^T = 0. \tag{2.26}$$

An important parameter of a codeword is its Hamming weight. For binary codes, this is the number of 1's in the codeword. The Hamming distance between two codewords is defined as the number of bit positions in which the two codewords differ. The smallest Hamming distance between any two distinct codewords is called the minimum distance, d_{min}, of the code. For linear codes, the minimum distance of a code is given by the minimum Hamming weight of any non-zero codewords in the code [33]. A (n, k) code with minimum distance d_{min} is called a (n, k, d_{min}) code.

For the Hamming code in equation (2.21), it can easily be shown that $d_{min} = 3$. Any two codewords in (2.22) differ in at least 3 bit positions. Therefore, this is a (7,4,3) Hamming code. The number of errors that can be corrected by a (n,k,d_{min}) code is $t \leq \frac{d_{min}-1}{2}$ [33]. Hence, the Hamming code can correct $t = 1$ error.

2.4.2 FEC Decoding

The transmitted codewords may be corrupted by multipath fading during transmission and by receiver noise. The FEC decoder then searches through all the possible codewords and selects the codeword that is closest to the received vector $\boldsymbol{r}$. This brute force search achieves ML decoding for any code. ML decoding is optimal in the sense that it minimizes the codeword error probability. The distance between the codewords and $\boldsymbol{r}$ may be measured using Hamming distance or Euclidean distance, resulting in hard or soft brute force searches, respectively. A (n, k) FEC code requires a search through 2^k codewords. This exponential growth becomes computationally prohibitive as k increases.

Fig. 2.5 shows the BER performance of uncoded and coded BPSK in AWGN, using the simple (7,4) Hamming code. The Hamming code is ML decoded using hard and soft brute force searches. First, we focus on the hard ML search. The Hamming code has $d_{min} = 3$ and can correct any codewords containing 1 bit error. At low SNR, the noise level is high and the codewords often contain more than 1 bit error. Therefore, the error-correcting capability is insufficient here, resulting

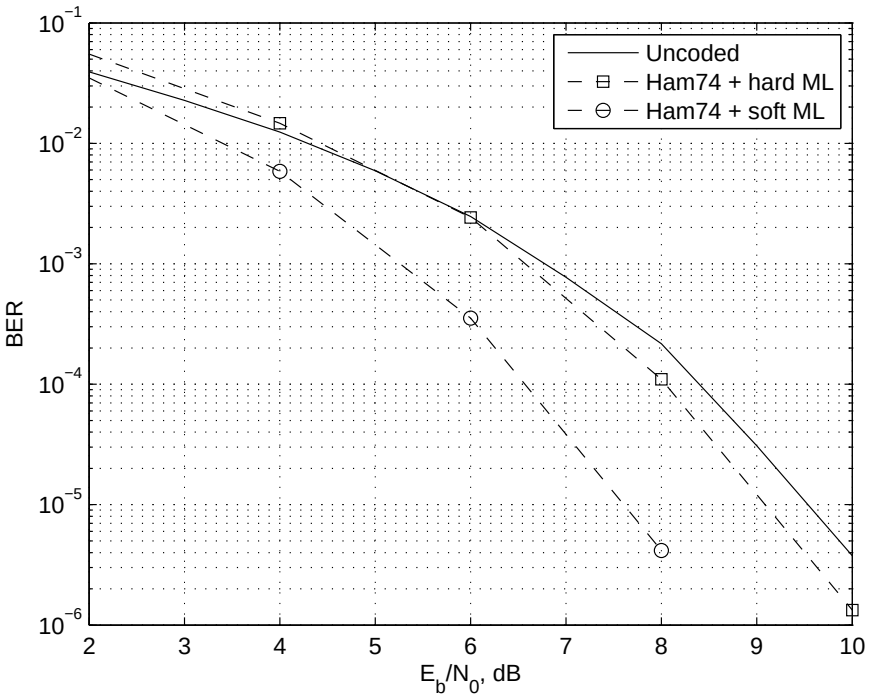


Figure 2.5: BER performance for uncoded and coded BPSK in AWGN using the (7,4,3) Hamming code (Ham74). The Hamming code is decoded using hard and soft brute force searches.

in performance degradation compared to an uncoded system. This is because the codeword bits have less energy compared to the uncoded information bits. Energy is normalized according to the number of information bits. As SNR increases, the probability that a codeword contains more than 1 bit error decreases. At high SNR, the Hamming code is able to correct all codewords with 1 bit error, whereas the same error in the uncoded system remains uncorrected. This results in a coding gain of 0.4dB over the uncoded system at BER=10^{-5}.

In hard decision brute force decoding, the output from the demodulator is quantized into 1's and 0's, resulting in a loss of information for the decoder. If the unquantized demodulator output is passed to the decoder, soft decision decoding based on minimum Euclidean distance is performed. As shown in Fig. 2.5, this results in a 1.9dB coding gain over the uncoded system at BER=10^{-5}. In most practical situations, significantly more powerful codes are used.

2.5 Summary

In this chapter, we have presented OSTBCs with $R_{stbc} \leq 1$ for two and four transmit antennas and QOSTBCs with $R_{stbc} \geq 1$ for four transmit antennas. Table 2.1 shows a summary of the properties of STBCs used in this thesis. QOSTBCs in general have higher R_{stbc} compared to OSTBCs, and so are used to achieve higher throughputs.

	Alamouti OSTBC	Tarokh OSTBC	Rate 1 QOSTBC	Rate 2 QOSTBC
n_T	2	4	4	4
R_{stbc}	1	1/2	1	2

Table 2.1
STBC properties.

Linear processing of OSTBCs decouples the transmitted symbols and is shown to achieve ML decoding. Linear processing of QOSTBCs generates symbol estimates with co-channel interference which degrades system performance. The effects of this interference can be compensated by using JD, SIC or PIC schemes. A detection complexity comparison between the three schemes is provided. It was shown that the SIC and PIC schemes have practical decoding complexity, compared to JD, as throughput is increased. For the work of this thesis, PIC is chosen over SIC because it has lower processing delay. PIC can also easily be implemented in hardware since digital signal processors (DSPs) and field-programmable gate arrays (FPGAs) exhibits parallel architectures.

Performance may be further improved by using FEC codes, the properties of which are summarized in Section 2.4. In Chapter 3, simple Hamming codes are used. More powerful LDPC codes are designed in Chapter 4 and used in Chapters 5, 6, 7 to obtain very good performance with relatively low decoding complexity.

Chapter 3

BMCM-OSTBC Structures

3.1 Introduction

Bit-interleaved coded modulation (BICM) [34, 35] is a well known structure for achieving both power and spectral efficiency. This structure can be serially concatenated with spatial multiplexing [36] or space-time codes [37] for transmission over multiple transmit antennas. In this chapter, we develop a coded modulation structure called interleaved bit-mapped coded modulation (IBMCM). For a constellation size of 2^M, this structure is essentially similar to using M bit-interleaved coded modulation (BICM) [34, 35] structures in parallel. A direct comparison between BICM and IBMCM is provided in Appendix B. To exploit the spatial diversity of multiple transmit antennas, the IBMCM structure is serially concatenated with orthogonal space-time block codes (OSTBCs). We focus here on the Alamouti OSTBC [8] for two transmit antennas and one receive antenna. This IBMCM-OSTBC structure may also be extended to more than two transmit antennas or to quasi-orthogonal space-time block codes. Very simple Hamming codes are used as the FEC component codes of the IBMCM and we show that, given sufficient interleaving depth, SNR gains of about 9dB over uncoded systems can be achieved at a bit error rate of 10^{-4}.

In Section 3.2, we describe the system and channel models used in this work. We assume quasi-static fading where the fading coefficients remain constant over a block of time and vary independently from block to block. Depending on the length of this block and interleaver depth, the FEC decoder will "see" two types of channel:

bit-static or codeword-static. Analytical bounds on the maximum likelihood (ML) performance of FEC block-coded MIMO systems, in bit-static and in codeword-static channels, are provided in Section 3.3. Section 3.4 presents performance results for this IBMCM-OSTBC structure using three different decoding schemes for the Hamming codes. The first is soft-decision brute force decoding which achieves ML performance, but is computationally infeasible for longer codes. The second is an iterative soft-input-soft-output (SISO) decoding algorithm [38], which is suitable for high-rate block codes. As a third scheme, we present the results for hard decision syndrome decoding of the block codes.

3.2 System Model

Fig. 3.1 depicts the system model. We describe the IBMCM-OSTBC scheme by means of an example employing QPSK modulation and simple FEC codes. We initially assume the use of very simple (7,4) Hamming codes, giving a system throughput of approximately 1.1 bits/symbol period. The model is readily extended to longer high rate codes. The encoder takes sixteen information bits and splits them into two streams of eight bits, one for in-phase transmission and the second for quadrature-phase transmission. Each stream of eight bits is separately encoded using the (7,4) Hamming code to produce two codewords of length seven. Let $\boldsymbol{C}_1^I = (C_1^I, C_2^I, ..., C_7^I)$ and $\boldsymbol{C}_2^I = (C_8^I, C_9^I, ..., C_{14}^I)$ be the two codewords for the in-phase (I) component and $\boldsymbol{C}_1^Q = (C_1^Q, C_2^Q, ..., C_7^Q)$ and $\boldsymbol{C}_2^Q = (C_8^Q, C_9^Q, ..., C_{14}^Q)$ be the two codewords for the quadrature-phase (Q) component, where $C_i^I, C_i^Q \in \{0, 1\}$ for $1 \leq i \leq 14$.

The two I (and Q) codewords are then separately interleaved using a simple seven-by-two rectangular interleaver. We call this an interleaver of depth two. This produces a stream of fourteen interleaved bits $C_1^I, C_8^I, C_2^I, C_9^I, ..., C_7^I, C_{14}^I$ (and $C_1^Q, C_8^Q, C_2^Q, C_9^Q, ..., C_7^Q, C_{14}^Q$).

The i^{th} bit from each stream of interleaved bits are used to select a QPSK symbol s_i, such that $s_i = [C_i^I, C_i^Q]$, which produces a Gray mapping. This results in a block of fourteen QPSK symbols $s_1, s_8, s_2, s_9, ..., s_7, s_{14}$, which are transmitted using the Alamouti OSTBC of [8], defined by equation (2.1). The Alamouti OSTBC takes

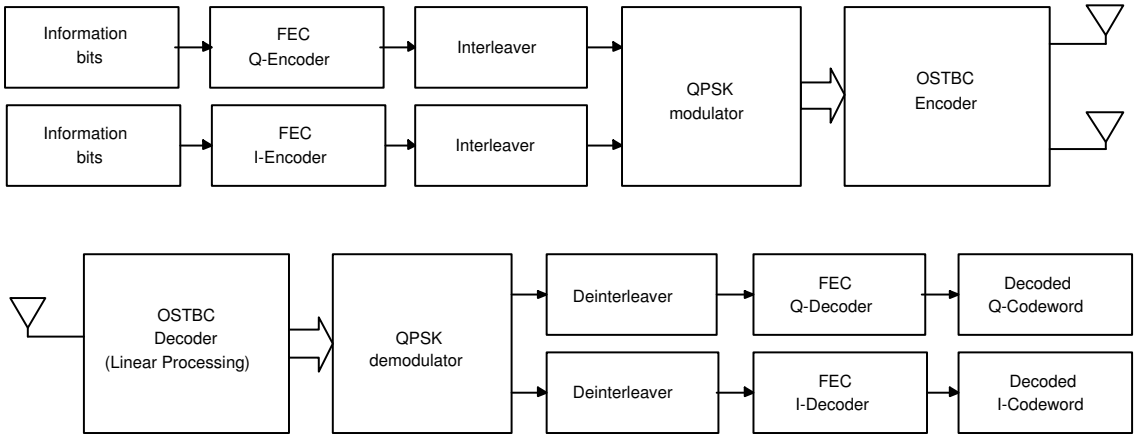


Figure 3.1: IBMCM-OSTBC structure for the encoding and decoding processes between FEC component codes and the Alamouti OSTBC [8].

pairs of QPSK symbols and produces four "space-time coded" symbols spanning two transmit antennas and two time slots. One receive antenna is used.

The stream of Alamouti OSTBC codewords is transmitted over the fading channel. The total transmitted energy is one unit of energy per information bit. Let $x_p(t)$ denote the symbol sent from transmit antenna p at time slot t and $\alpha_1(t)$ and $\alpha_2(t)$ denote the complex Gaussian fading coefficients affecting $x_1(t)$ and $x_2(t)$, respectively. The OSTBC scheme requires that the fading coefficients remain fixed over each two-symbol OSTBC block ($L = 2$), i.e. $\alpha_1(t+1)=\alpha_1(t)$ and $\alpha_2(t+1)=\alpha_2(t)$. The faded symbols from both transmit antennas superimpose to form the received signal

$$r(t) = \sum_{p=1}^{2} \alpha_p(t)x_p(t) + w(t), \qquad t = 1, 2 \tag{3.1}$$

where $w(t)$ is the additive white Gaussian (AWGN) noise at the receive antenna. The received signals are passed through an OSTBC decoder, where linear processing is performed to decouple the transmitted symbols from each antenna. For the Alamouti OSTBC, linear processing is given by equation (2.8) and the resulting symbol estimates are given by equation (2.9). For any OSTBC, linear processing at the receiver achieves ML decoding [24]. Perfect channel state information is required at the receiver to achieve this [8].

The soft outputs of the OSTBC decoder are demapped and deinterleaved into the respective I and Q soft codewords. These are then passed to the (7,4) Hamming

decoder to determine the most likely information bits transmitted. Three different FEC decoding algorithms are considered for the Hamming codes in Section 3.4.

3.2.1 Channel Model: Bit-Static Channel

An OSTBC requires L time slots to transmit all its symbols. The quasi-static fading assumption requires that the fading coefficients remain constant over each block of L time slots and vary independently from block to block. This is a requirement for ideal use of OSTBCs [8] as it allows the transmitted symbols to be completely decoupled from each other at the receiver using perfect CSI and simple linear processing.

Two different quasi-static fading channel models, as seen by the FEC decoder, are investigated here. The first is a bit-static fading channel, where every bit within the deinterleaved I and Q codewords experiences independent fading coefficients. This is achieved by using a block interleaver of depth γ, which is set to L. For the Alamouti OSTBC, $L = 2$ time slots. Fig. 3.2 shows how the simple seven-by-two block interleaver breaks up the quasi-static fading experienced by two consecutive transmitted symbols to achieve this. Fig. 3.2a shows the two codewords in the I-component. These codeword bits are interleaved with each other and then mapped with the corresponding bits from the Q-component to produce the stream of fourteen QPSK symbols in Fig. 3.2b. The symbols s_1 and s_8 in the first two time slots (far right) are encoded using an OSTBC and they experience the fading coefficients $\alpha_1(1)$ for transmit antenna 1 and $\alpha_2(1)$ for transmit antenna 2. Note that $\alpha_1(1) = \alpha_1(2)$ and $\alpha_2(1) = \alpha_2(2)$. Fig. 3.2c shows the I-component codeword, as seen by the FEC decoder after the demapping and deinterleaving process. The deinterleaving process is responsible for breaking up the quasi-static fading required by the Alamouti OSTBC. As a result, the FEC decoder sees 7 independent fading coefficients, compared to only 4 if no interleaving/deinterleaving were to be used. It has been shown that for any normalized fade rate $f_D T$, the required distance between two symbols for them to essentially fade independently is [39]

$$\beta > \frac{0.3}{f_D T} \qquad \text{time slots.} \tag{3.2}$$

For example, when $f_D T = 0.01$, to achieve independent fading between two symbols in the transmitted sequence, there needs to be a spacing in time of at least $\beta =$

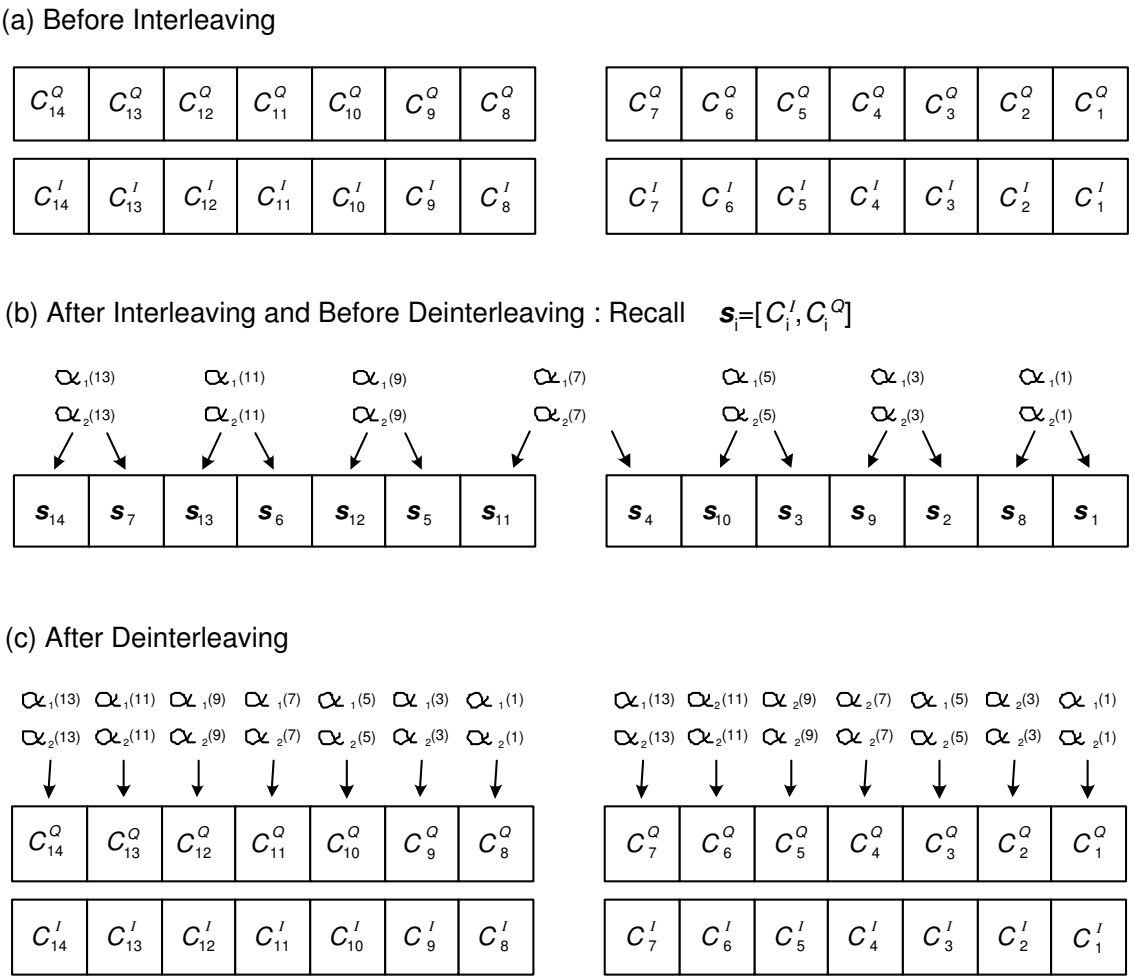


Figure 3.2: How the FEC decoder sees a bit-static fading channel.

30 time slots. To achieve a bit-static channel in this case, we would need a block interleaver of depth $\gamma = \beta$. A large interleaver depth introduces a large delay time.

From equation (3.2), it is clear that we need $f_D T > 0.15$ to meet the quasi-static requirement of the Alamouti OSTBC ($\beta = 2$). At the high data rates typical of space-time codes, this represents very fast fading. For example, at a carrier frequency of 2 GHz and a data rate of 2 Mb/s, the receiver would have to be travelling at speeds in excess of 162 000 km/h.

3.2.2 Channel Model: Codeword-Static Channel

For a more realistic scenario, we need to lower $f_D T$ which has the effect of slowing down the time variation of the channel. This allows the FEC decoder to see a

codeword-static channel, where the fading coefficients remain constant over an entire codeword instead of only over each bit. An interleaver of depth $\gamma = L$ spreads each codeword out across nL time slots, where n is the block length of the FEC codeword, and L is the minimum number of required time slots for the fading to remain constant for ideal use of OSTBCs. To see a codeword-static channel, the fading coefficients now need to remain constant for every block of nL time slots. For the simple (7,4) Hamming code and the Alamouti OSTBC, this is achieved by fixing the fading coefficients during transmission to be constant for fourteen consecutive time slots, i.e.

$$\alpha_1(t+13) = \alpha_1(t+12) = ... = \alpha_1(t)$$
$$\alpha_2(t+13) = \alpha_2(t+12) = ... = \alpha_2(t).$$

Note that this is independent of the quasi-static assumption for the Alamouti OSTBC of [8]. The minimum requirement for this OSTBC is satisfied if fading is constant for an even number of time slots. An interleaver of depth 2 becomes redundant in the quasi-static channel as both codewords used in the interleaving process experience the same fading.

3.3 Analytical Upper Bound for ML Decoding

In this section, we develop an upper bound on the ML performance of FEC coded MIMO systems. This bound is a block code adaptation of the one derived in [40] for convolutional codes in AWGN channels, which may be expressed as

$$P_b < \sum_{d=d_{free}}^{\infty} c_d P_d \tag{3.3}$$

where P_b is the bit error probability, P_d is the probability of an error event with Hamming weight d and d_{free} is the free distance of the convolutional code. The values of c_d can be obtained from the transfer function of the convolutional code [40, 41]. These values are essentially a product of the number of codewords with Hamming weight d and the number of bit errors associated with these codewords. The values of c_d are a measure of the average number of bit errors occurring each time an error event with Hamming weight d occurs.

The work of [42] generalized this bound to evaluate the performance of convolutional codes in MIMO systems for Rayleigh fading channels. It was shown that as the number of antennas in the system increases, the fading channel performance is shifted towards that of an AWGN channel. We modify this bound to evaluate the use of block codes in MIMO systems for the bit-static and codeword-static channels. Just as the transfer function of [40, 41] contains information regarding the weight distribution of the convolutional code, the weight enumerator function of [12] contains information regarding the weight distribution of block codes.

In order to find an upper bound for the bit error probability of an (n,k,d_{min}) block code, we need to relate the pairwise probability associated with an error event of weight d to the bit error probability. This is done by the bit error multiplicity [34, 12] of the weight enumerator function given by

$$b_d = \sum_w \frac{w}{k} A_{w,d}. \tag{3.4}$$

where $A_{w,d}$ is the number of codewords of Hamming weight d with information weight w.

By replacing the c_d term in equation (3.3) with equation (3.4), the bit error probability for a block FEC coded system using ML soft decoding can be upper bounded by

$$P_b < \sum_{d=d_{min}}^{\infty} b_d P_d. \tag{3.5}$$

3.3.1 Union Bound for Bit-Static Channels

In our example, a fully-independent channel can be achieved by using a simple seven-by-two block interleaver, as shown in Fig. 3.2. In general, assuming an infinitely large interleaver depth so that a bit-static channel for M-PSK transmission is achieved, P_d from equation (3.5) can be shown to be [42]

$$P_d \approx \frac{1}{\log_2 M} \left[1 - \mu_d \sum_{k=0}^{dn_T n_R - 1} \binom{2k}{k} \left(\frac{1-\mu_d^2}{4} \right)^k \right] \tag{3.6}$$

where

$$\mu_d = \sqrt{\frac{1}{1 + \frac{n_T N_0}{R_c E_b \log_2(M) sin^2\left(\frac{\pi}{M}\right)}}} \tag{3.7}$$

and R_c is the code rate of the FEC code. The number of transmit and receive antennas are represented by n_T and n_R, respectively. E_b/N_0 is the post-processing SNR, instead of the SNR per receive antenna. A discussion of these SNR definitions is provided in Appendix A. The d term in the summation in equation (3.6) shows that the FEC code provides diversity gain in bit-static channels.

3.3.2 Union Bound for Codeword-Static Channels

In the case of a codeword-static channel, we assume that every bit within the FEC codeword experiences the same fading, but the fading varies independently after every γ codewords. P_d can then be shown to be [42]

$$P_d \approx \frac{1}{\log_2 M}\left[1-\mu_d \sum_{k=0}^{n_T n_R-1}\binom{2k}{k}\left(\frac{1-\mu_d^2}{4}\right)^k\right] \tag{3.8}$$

where

$$\mu_d = \sqrt{\frac{1}{1+\frac{n_T N_0}{dR_c E_b \log_2(M) sin^2\left(\frac{\pi}{M}\right)}}} \tag{3.9}$$

Note that the d term disappears from the upper limit of the summation in equation (3.8), implying that no diversity gain is obtained from the FEC code in codeword-static channels.

3.3.3 Union Bound for Uncoded MIMO systems

When no FEC coding is used, this is equivalent to using a (1,1,1) code. Therefore, we set $d = 1$ in equation (3.6) and use the result in equation (3.5) to obtain a union bound on performance. Since the codeword length is now 1, the channel is by default bit-static and also codeword-static. We can show this by observing that (3.6) equals (3.8) and (3.7) equals (3.9) when $d = 1$.

3.4 Performance Results

We now present simulation results using three different decoding methods. At least 200 bit errors are simulated for each point. We use the very simple (7,4) Hamming code for soft-decision brute force decoding as it is small enough for implementation. We then show that the iterative SISO decoding algorithm of [38] can be used to

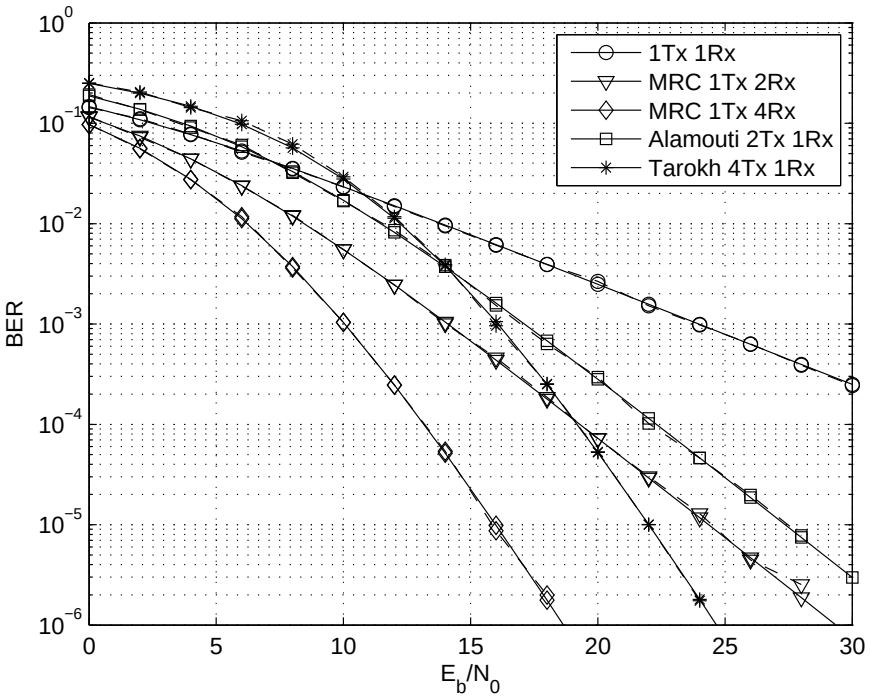


Figure 3.3: Comparison between the union bound (solid lines) and simulation results (dashed lines) for uncoded MIMO systems with no FEC coding.

decode longer codes, such as the (15,11) Hamming code. Finally we compare these soft decision schemes against hard decision syndrome decoding using the same (7,4) Hamming codes. For two transmit antennas, we employ the Alamouti OSTBC [8] and for four transmit antennas, we use the Rate 1/2 Tarokh OSTBC [24]. Maximum ratio combining (MRC) is employed for multiple receive antennas. Perfect CSI is available at the receiver.

3.4.1 Uncoded MIMO systems

When no FEC coding is used, we set $d_{min} = 1$ in equation (3.5). Fig. 3.3 shows the performance of uncoded MIMO systems with different numbers of transmit and receive antennas. The simulation results are shown as dashed lines and the corresponding Union Bounds are shown as solid lines. It is clear from Fig. 3.3 that the bound is very tight across all observed SNR for uncoded MIMO systems.

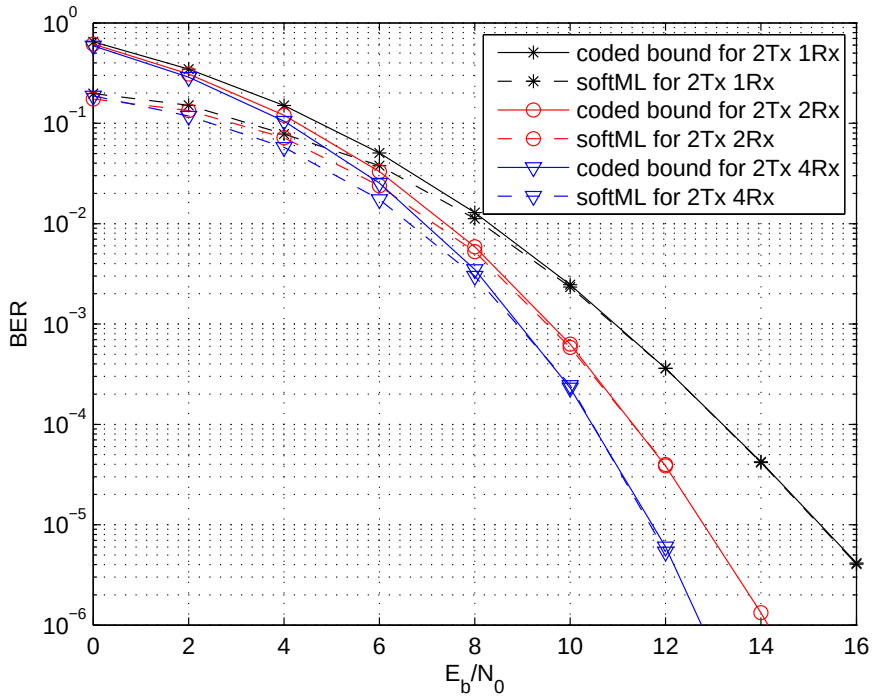


Figure 3.4: BER performance comparison between brute force (soft ML) decoding and the union bound in a bit-static channel for $n_T = 2$ and $n_R = 1, 2, 4$.

3.4.2 Brute force decoding

Brute force ML decoding searches through all possible codewords and chooses the codeword with minimum Euclidean distance. Throughout this section, we use the (7,4) Hamming code to compare the performance of brute force decoding against the ML bounds from Section 3.3. Fig. 3.4 compares the coded ML performance using brute force decoding against the upper bound in equation (3.5) for a system with two transmit and one, two or four receive antennas in a bit-static fading channel. Since this is a Union Bound, the bound is only tight at high SNR and diverges for low SNR.

Fig. 3.5 shows that in a codeword-static fading channel, the bound is loose for a two transmit and one receive antenna system, but becomes tighter as the number of receive antennas increases. As this is a Union bound, we know that the bound is loose at low SNR i.e. when the channel is noisy. As more receive antennas are added, the probability of all subchannels having deep fades decreases so the channel effectively becomes less noisy. Hence the bound becomes tighter with an increasing

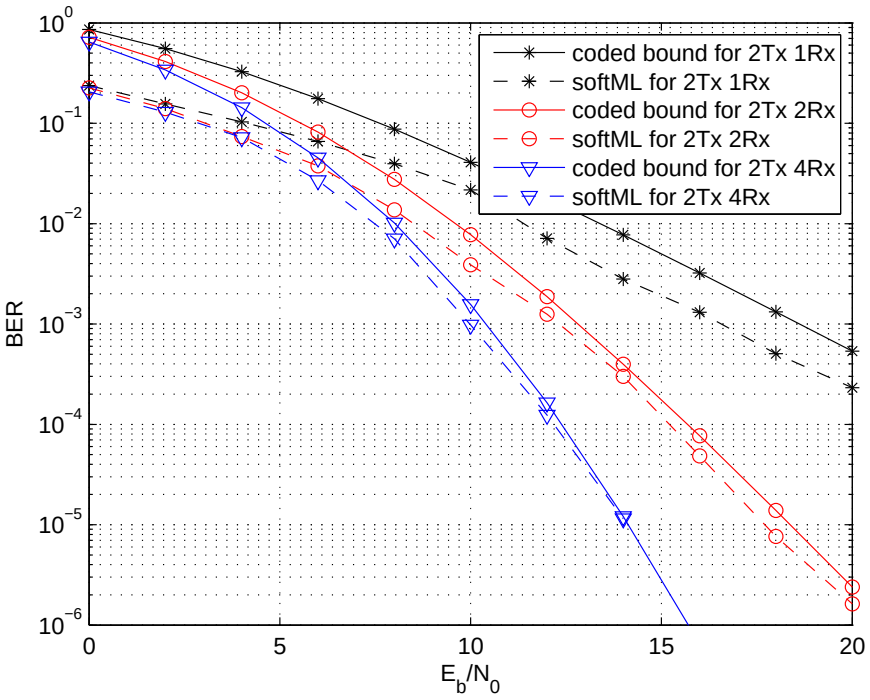


Figure 3.5: BER performance comparison between brute force (soft ML) decoding and the union bound in a codeword-static channel for $n_T = 2$ and $n_R = 1, 2, 4$.

number of receive antennas.

Next, we compare the performance of uncoded and coded systems in a bit-static fading channel, as shown in Fig. 3.6. At a BER of 10^{-4}, the coded system achieves a total SNR gain of about 9dB over the uncoded system for one receive antenna. Much of this gain is because of the increased diversity provided by the Hamming code structure. For two and four receive antennas, the total SNR gain over uncoded systems due to both diversity and coding is only 5dB and 3dB, respectively. By adding more receive antennas to a system, the fading channel is effectively shifted towards an AWGN channel [42]. Note that the coding gain of the (7,4) Hamming code in an AWGN channel is about 1.7dB [38] at a BER of 10^{-4}.

Fig. 3.4 and Fig. 3.5 show that the SNR gain obtained by a system with $2N$ receive antennas over a system with N receive antennas decreases as N increases. This is because the translation of diversity gain (in decades/dB) into SNR gain (in dB) decreases as the slope of the BER curve increases. This helps to explain the trend of decreasing SNR gain in Fig. 3.6 of coded systems compared to uncoded

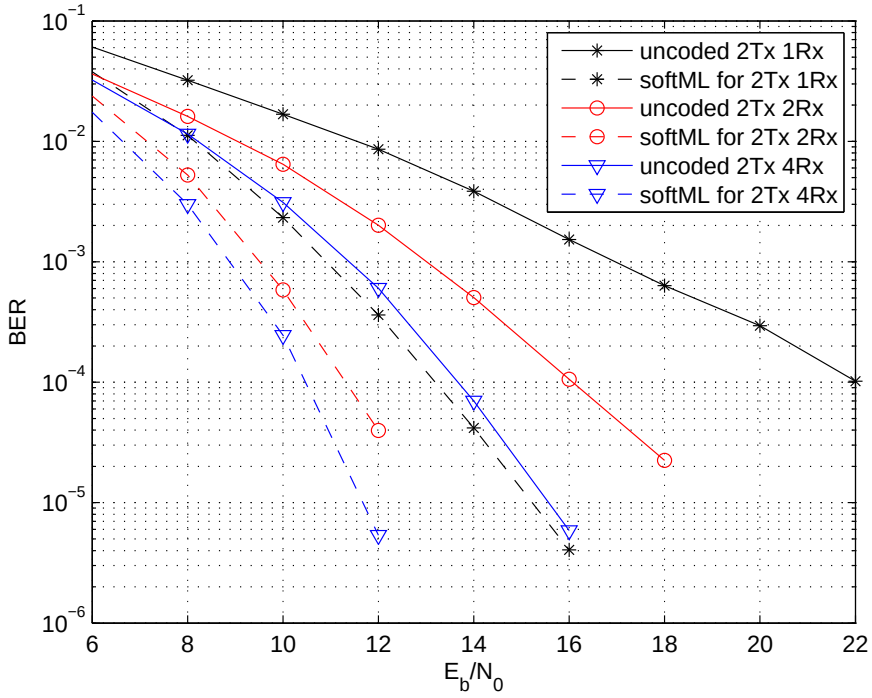


Figure 3.6: BER performance comparison between uncoded and coded systems using brute force (soft ML) decoding in a bit-static channel for $n_T = 2$ and $n_R = 1, 2, 4$.

systems as the diversity is increased by adding more receive antennas.

Fig. 3.7 shows the comparative performance of uncoded and coded systems in a codeword-static fading channel. The performance curves for each coded system have almost the same slope as the curves for the corresponding uncoded system. This clearly indicates that little or no additional diversity gain is provided by the Hamming code when the Hamming decoder sees the same fading coefficient across all codeword bits[1]. This is also confirmed by the limit in the summation of equation (3.8). In a codeword-static channel, diversity gain is determined only by the number of transmit and receive antennas. At a BER of 10^{-4}, the coded system achieves an SNR gain of only 0.4dB over the uncoded system for one receive antenna. Most of this gain is due to the coding gain from the Hamming code. As the number of receive antennas is increased from two to four, the SNR gain increases to 0.8dB and 1.5dB, respectively. This trend of increasing SNR gain is because as more receive

[1]For the proposed IBMCM-STBC system with interleaver depth γ, we fix the channel to remain constant over $n\gamma$ time slots.

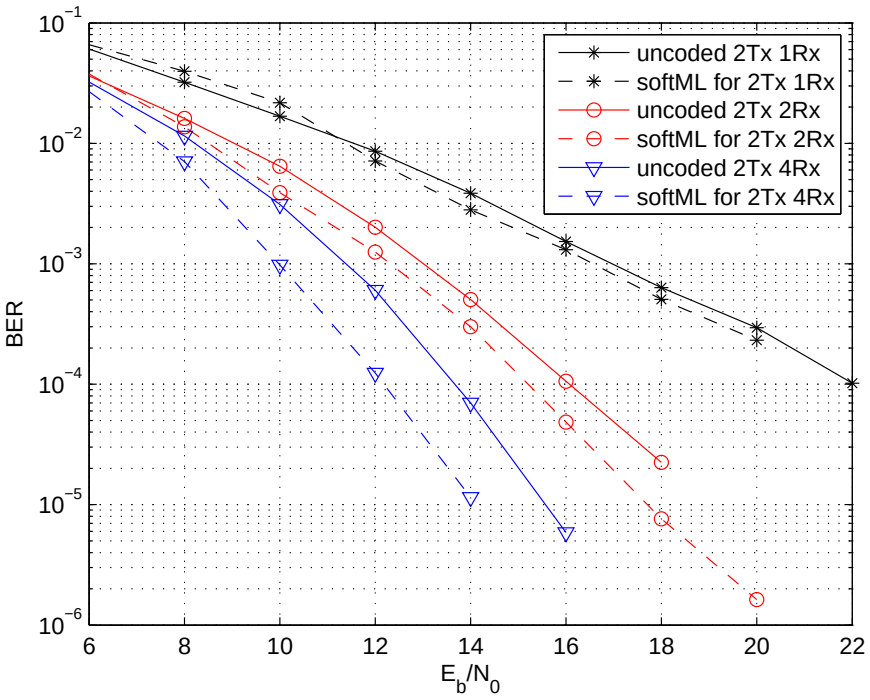


Figure 3.7: BER performance comparison between uncoded and coded systems using brute force (soft ML) decoding in a codeword-static channel for $n_T = 2$ and $n_R = 1, 2, 4$.

antennas are added, the probability that every receive antenna experiences a deep fade simultaneously decreases and the channel effectively becomes closer to being an AWGN channel.

3.4.3 SISO Decoding

Although soft decision brute force decoding over all codewords achieves ML performance, this becomes computationally infeasible as the code length increases. A practical solution for high-rate codes is to use sub-optimal, iterative SISO decoding [38] of the FEC code.

At the receiver, the received symbols are combined in the STBC decoder and the log-likelihood ratio (LLR) is calculated for every received bit for both the I and Q components. These LLRs are then utilized in the FEC decoders of Fig. 3.1.

Systematic binary block codes can be modelled [38] as a combination of parallel concatenated single parity-check (SPC) codes which can be jointly and iteratively

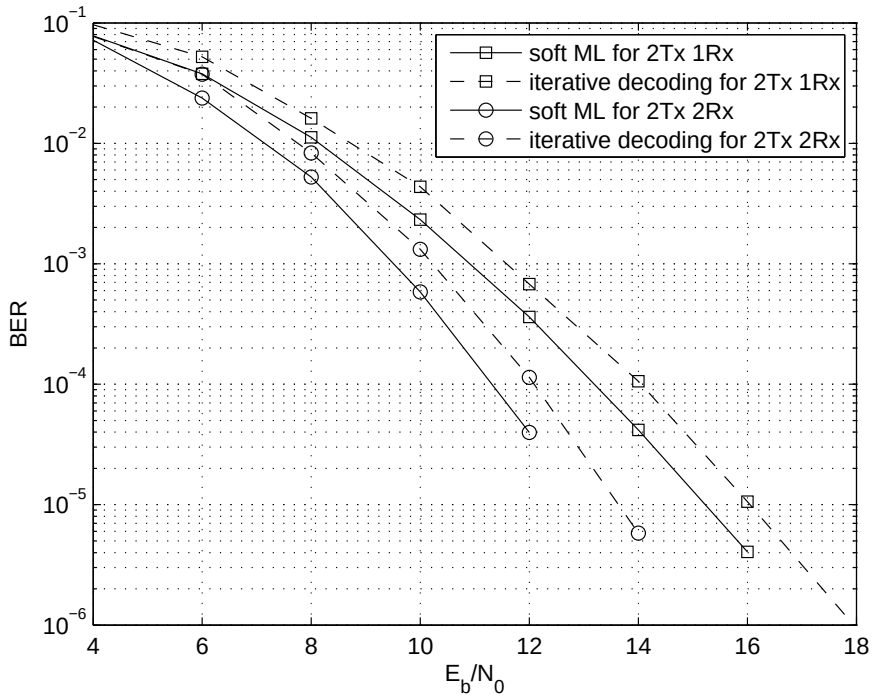


Figure 3.8: BER performance comparison between brute force decoding and iterative SISO decoding in a bit-static fading channel for $n_T = 2$ and $n_R = 1, 2$. (7,4) Hamming codes are used.

decoded. During each iteration of the SISO decoder, the extrinsic information generated for the data bits by each SPC decoder is shared with other SPC decoders that utilize those particular data bits. After the last iteration, the extrinsic information for a given data bit is summed and hard decisions are made.

Fig. 3.8 shows coded performance of a two-transmit, one-receive antenna system [8] using brute force decoding and using SISO decoding in a bit-static channel. The performance for two receive antennas is also shown. For the (7,4) Hamming code, the SISO algorithm converges after 3 iterations. In both cases, the sub-optimality of the SISO decoding algorithm is only about 0.8dB.

Fig. 3.9 compares the performance of the (7,4) Hamming code and the higher rate (15,11) Hamming code for two transmit and one receive antennas using SISO decoding in a bit-static fading channel. The SISO algorithm for the (15,11) Hamming code converges after 5 iterations. The union bounds for both codes are also displayed. All the curves have the same slopes, implying that SISO decoding is able to achieve the same diversity gain when using the (15,11) Hamming code as when the (7,4)

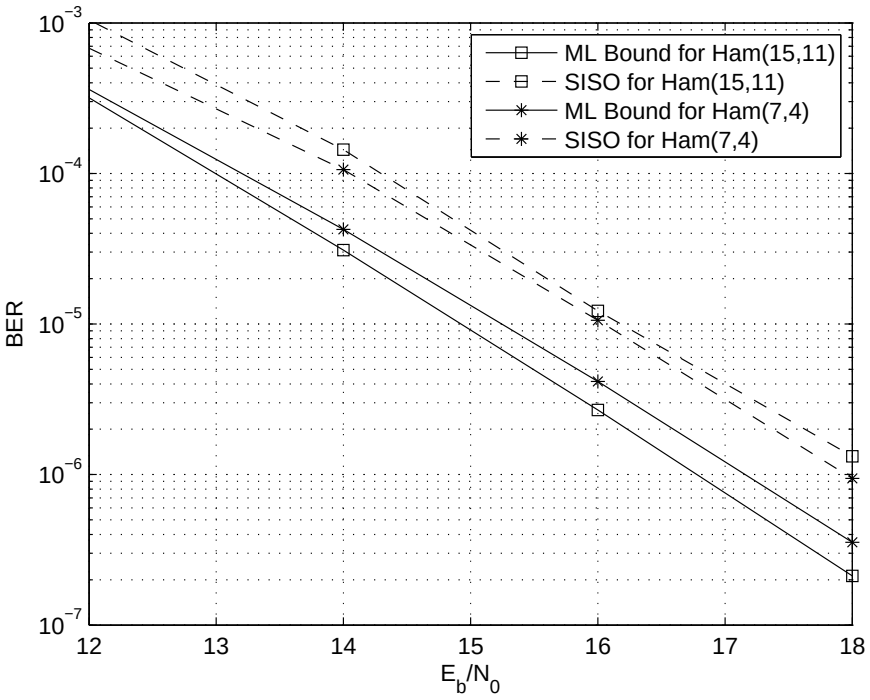


Figure 3.9: BER performance comparison for the (7,4) and (15,11) Hamming codes using SISO decoding, together with their union bounds, in a bit-static fading channel for $n_T = 2$ and $n_R = 1$.

Hamming code is used. There is only a small SNR loss of about 0.2dB by using the Rate 0.73 (15,11) Hamming code, compared to that using the Rate 0.54 (7,4) Hamming code.

A comparison of the ML upper bounds in Fig. 3.9 indicates that the higher rate (15,11) Hamming code in fact performs about 0.4dB better than the (7,4) Hamming code at high SNR, if ML decoding is used. This implies that the sub-optimality of the SISO decoding algorithm costs about 1.4dB for the (15,11) Hamming code.

3.4.4 Hard Syndrome Decoding

The SISO decoding method passed soft information to the FEC decoder and hard decisions were made at its output. In this subsection, we make hard decisions at the QPSK demodulator output of Fig. 3.1. Hard decision syndrome decoding [12] is used in the FEC decoder. This is also referred to as algebraic decoding and it has much lower complexity.

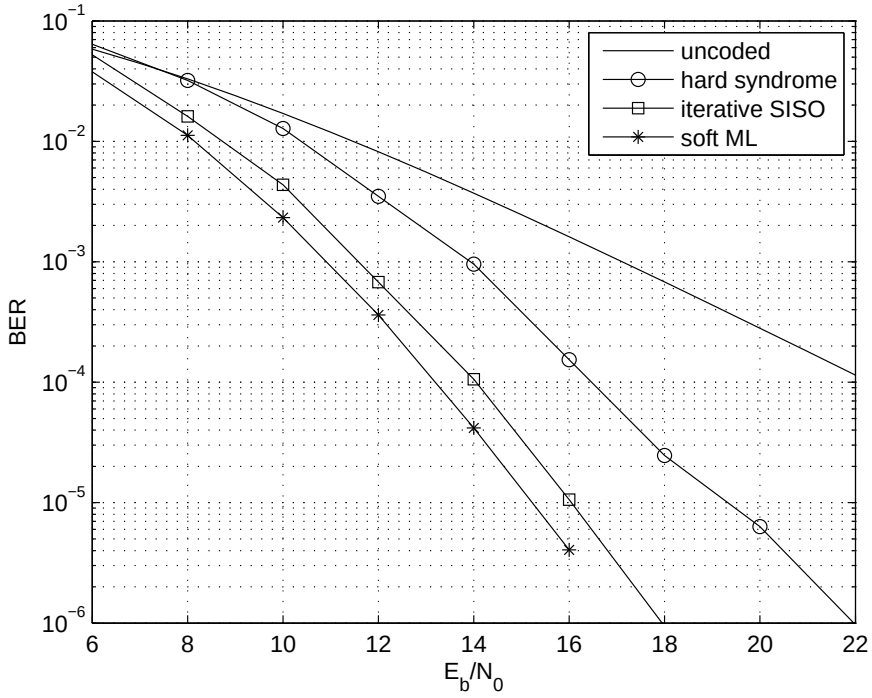


Figure 3.10: BER performance comparison for (7,4) Hamming code using hard syndrome, SISO and brute force decoding, in a bit-static fading channel for $n_T = 2$ and $n_R = 1$.

Fig. 3.10 compares the performance of the (7,4) Hamming code using hard decision syndrome decoding, iterative SISO decoding and brute force (soft ML) decoding in a bit-static fading channel. The slope of the performance curve using hard syndrome decoding is not as steep as when using soft decisions, so there appears to be some loss in diversity gain when hard decisions are used. At a BER of 10^{-4}, the hard syndrome decoding provides an SNR gain of only about 5.5dB over the uncoded system, as compared to 8dB using iterative SISO decoding. A further 0.8dB gain is provided by brute force decoding at the same BER.

3.5 Summary

In this chapter, we have presented a simple serially concatenated structure consisting of the OSTBC of [8] and IBMCM with simple Hamming component codes. It was found that despite the simplicity of the FEC component codes used, significant diversity and coding gains were achieved in a bit-static fading channel, compared

to systems with no FEC coding. The amount of fading variation seen by the FEC decoder is an important parameter. No diversity gain is obtained when using the FEC codes in a codeword-static fading channel, but performance can be improved by increasing the number of receive antennas. An analytical bound for ML performance was also presented to verify our simulation results.

Performance can be improved by using more powerful low density parity check codes, which will be developed in Chapter 4. The use of OSTBCs in this chapter restricts system throughput. In Chapter 5, we show significantly higher system throughput can be achieved by using quasi-orthogonal space-time block codes.

Chapter 4

Low Density Parity Check Codes

4.1 The Renaissance of LDPC Codes

LDPC codes were first introduced by Gallager [43] in 1960. However, they were largely ignored due to the lack of computing power at the time and eventually forgotten. The discovery of Turbo codes in 1993 by Berrou et. al. [44] provided the coding community with a new iterative decoding technique that enabled performance approaching the Shannon limit[1] [9]. This led to the rediscovery of LDPC codes by Mackay and Neal [45] in 1995. Although the performances of both Turbo codes and LDPC codes can approach the Shannon limit, LDPC codes have some advantages over Turbo codes [46]: (1) They do not require a long interleaver to achieve good error performance; (2) They have better block error performance; (3) Their error floor occurs at a much lower BER; and (4) Their decoding is not trellis based.

LDPC codes can be decoded using a low complexity belief propagation (BP) algorithm and still achieve performance very close to that using maximum-likehood (ML) decoding. Simulations for an LDPC code of length 10^7 have shown that performance within 0.04dB of the Shannon limit can be achieved at a bit error rate of 10^{-6} [26]. The caveat is that BP decoding performs significantly worse in the presence of short cycles in the parity check matrix, especially length 4-cycles [47, 48, 49].

There are many ways to construct LDPC codes. In [43], Gallager proposed a

[1]The Shannon limit is the SNR at which error free performance is possible asymptotically with increasing block length.

method to generate pseudorandom LDPC codes. Good LDPC codes can be found using computer searches. However, these codes usually have long block lengths and their encoding is complex due to the lack of structure. In [50], a heuristic method called "bit-filling" is used to search for LDPC codes with large girth. However, this does not guarantee that the code will have the largest possible girth, for a given block length, n. For $n < 10,000$, there is another heuristic algorithm [51] which searches for a good LDPC code based on the average girth distribution of the code.

The first algebraic construction of LDPC codes based on finite geometries was introduced by Kou et. al. [52] in 2000. Finite geometry codes have girth of at least 6 and they can be linearly time-encoded using simple feedback shift registers. More recently, LDPC codes have also been constructed based on combinatorial designs [53] known as balanced incomplete block designs. An overview of the various construction methods is provided in [54].

In Section 4.2, we show how the smallest regular matrix can be constructed, given the required girth, column and row weights. In Section 4.3, we develop a simple LDPC code construction which guarantees a girth of at least 6. This construction method also creates codes with flexible[2] rates for the same block length. The properties of these LDPC codes are presented in Section 4.4. In Section 4.5, we show their performance in an additive white Gaussian noise (AWGN) channel, and we compare them to other existing LDPC codes. In Section 4.6, the relationship between the proposed LDPC codes and other FEC codes are highlighted.

4.2 Finding the smallest regular matrix

Most LDPC codes with large girth, and found by computer search, have long block lengths. This introduces a long delay as the receiver needs to wait for the whole codeword to be detected before decoding can commence. This motivated us to find the smallest regular $\boldsymbol{H}$ matrix for any given girth ζ, column weight q and row weight p. A regular matrix have constant row and column weights.

Fig. 4.1 shows the $\boldsymbol{H}$ matrix associated with several girths ζ for the special case when $p = q = 2$. The rows and columns of $\boldsymbol{H}$ are represented in the bipartite

[2]The advantages of this rate flexibility are explained and utilized in Chapters 5 and 6.

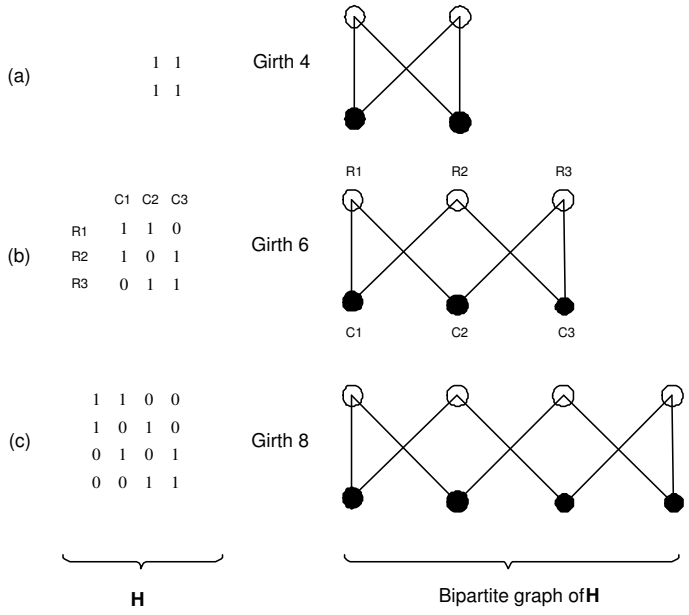


Figure 4.1: The smallest $\boldsymbol{H}$ matrices with girths ζ=4,6,8, for fixed $p = q = 2$ are shown along with their corresponding birpartite graphs.

graphs by the empty and filled circles, respectively. An edge joins the i^{th} empty circle to the j^{th} filled circle if the element in the i^{th} row and j^{th} column of $\boldsymbol{H}$ is a "1". For example, in Fig. 4.1b, row R1 contains a "1" in columns C1 and C2. In the corresponding bipartite graph, this is represented by an edge connecting R1 to C1 and C2.

A cycle in a bipartite graph is determined by the number of edges required to connect a vertex (in this case, either an empty or filled circle) back to itself, without using the same edge or vertex more than once. The smallest cycle in the graph is then called the girth ζ. The bipartite graphs of the $\boldsymbol{H}$ matrices in Fig. 4.1 are defined by a single cycle of girth ζ. Therefore, these $\boldsymbol{H}$ matrices are clearly the smallest matrices possible given a girth ζ, when $p = q = 2$.

For $p = q$ values greater than 2, finding the smallest matrix for a given girth is non-trivial. The rest of this section is dedicated to solving this problem using a two-stage process. In the first stage, part of the $\boldsymbol{H}$ matrix, denoted $\boldsymbol{Y}$, is constructed in order to guarantee it is the smallest matrix possible for a given girth ζ. In the second stage, we must try to find a smaller regular matrix $\boldsymbol{Z}$, with girth at least ζ and column and row weights of $p - 1$, to fill the remaining part of $\boldsymbol{Y}$.

4.2.1 Stage 1: Guarantee of smallest matrix possible

We now describe the algorithm used to find the smallest matrix for a specified girth ζ, row weight p and column weight q, given that[3] $p = q$. We use the example of $p = q = 3$ to illustrate the algorithm in Fig. 4.2.

(Step 1A) First we create a blank p-by-p matrix $\boldsymbol{Y}$. Then we fill the elements in the topmost row of $\boldsymbol{Y}$ with "1"s. We also fill the elements in the leftmost column of $\boldsymbol{Y}$ with "1"s. The remaining unfilled elements form a $(p-1)$-by-$(p-1)$ matrix $\boldsymbol{Z}$ (see Fig. 4.2). If we want $girth(\boldsymbol{Y}) = 4$, then we need to find a regular matrix Z that has both column and row weight $p-1$ and $girth(\boldsymbol{Z}) \geq 4$. If we want $girth(\boldsymbol{Y}) > 4$, then we fill all the remaining elements of $\boldsymbol{Y}$ with "0" and go to Step 1B.

(Step 1B) Next we extend $\boldsymbol{Y}$ to be a $[p + (p-1)^2]$-by-$[p + (p-1)^2]$ matrix. For the p topmost rows of the extended matrix $\boldsymbol{Y}$, we fill all the elements in these rows such that each row has weight p and no two newly filled "1"s occur in the same column. Similarly, for the p leftmost columns of the extended matrix $\boldsymbol{Y}$, we fill all the elements in these columns such that each column has weight p and no newly filled "1"s occur in the same row (See Fig. 4.2). If we want $girth(\boldsymbol{Y}) = 6$, then we need to find a regular $(p-1)^2$-by-$(p-1)^2$ matrix $\boldsymbol{Z}$ that has both column and row weight of $p-1$ and $girth(\boldsymbol{Z}) \geq 6$. If we want $girth(\boldsymbol{Y}) > 6$, then we fill all the remaining elements of $\boldsymbol{Y}$ with "0" and go to Step 1C.

(Step 1C) Next we extend $\boldsymbol{Y}$ to be a $[p+(p-1)^2+(p-1)^3]$-by-$[p+(p-1)^2+(p-1)^3]$ matrix. For the $p + (p-1)^2$ topmost rows of the extended matrix $\boldsymbol{Y}$, we fill all the elements in these rows such that each row has weight p and no two newly filled "1"s occur in the same column. Similarly, for the $p+(p-1)^2$ leftmost columns of the extended matrix $\boldsymbol{Y}$, we fill all the elements in these columns such that each column has weight p and no newly filled "1"s occur in the same row (See Fig. 4.2). If we want $girth(\boldsymbol{Y}) = 8$, then we need to find a regular $(p-1)^3$-by-$(p-1)^3$ matrix $\boldsymbol{Z}$ that has both column and row weight of $p-1$ and $girth(\boldsymbol{Z}) \geq 8$. If we want $girth(\boldsymbol{Y}) > 8$, then we fill all the remaining elements of $\boldsymbol{Y}$ with "0" and go to the next step[4].

This is an inductive process which can be extended infinitely. In each step

[3]This algorithm requires $p = q$ as the next step (Step 1A) requires a square regular matrix with row weight $(p-1)$ and column weight $(q-1)$. This is clearly impossible if $p \neq q$.

[4]Step 1D in this case.

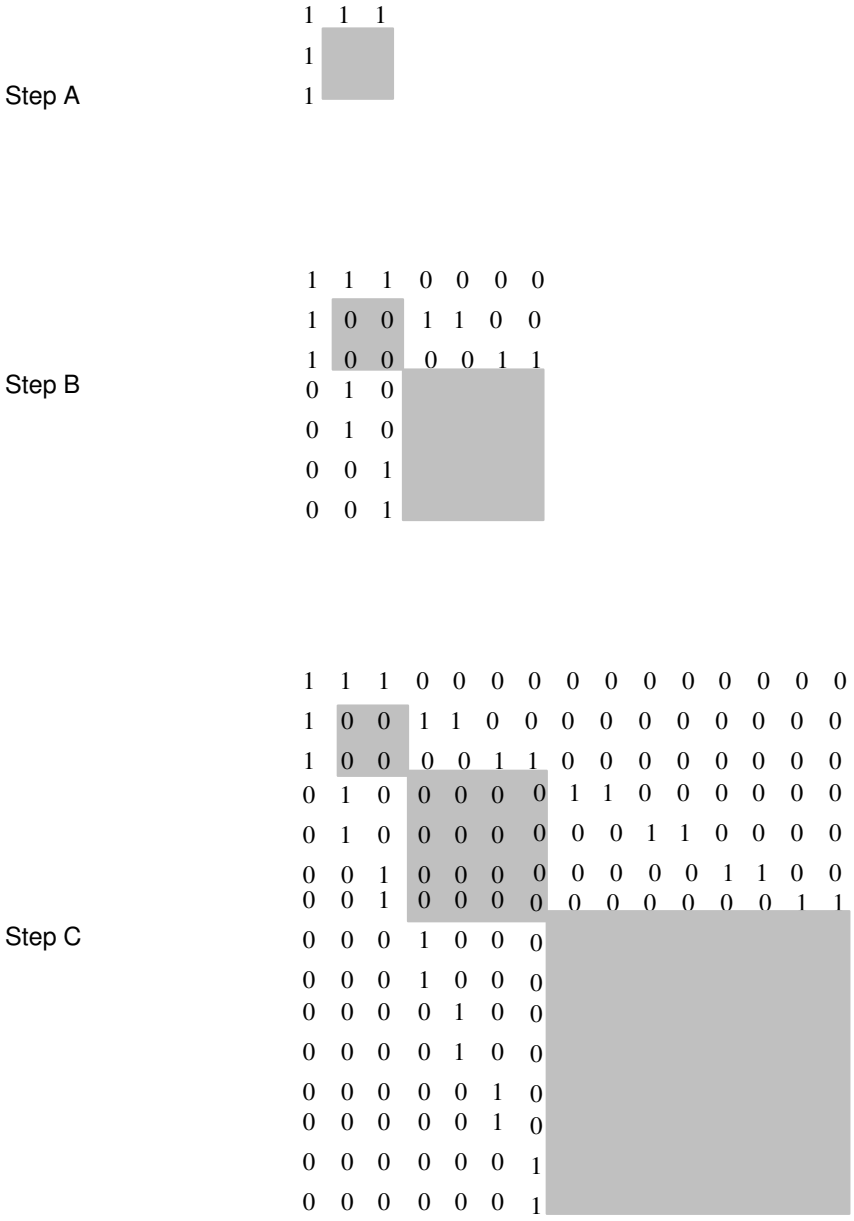


Figure 4.2: Example of Step 1 algorithm to find the smallest matrix for any given ζ and $p = q = 3$.

increment, we continue to extend the matrix $\boldsymbol{Y}$. The corresponding girth also increases by 2 with each step. This algorithm is similar to but different from that found in [55], which is used to find the girth of any given matrix. In Fig. 4.2, the elements "1" form a "V" structure rotated clockwise by 135 degrees, hence we call Step 1 the V-construction. This algorithm does not guarantee the existence of a matrix $\boldsymbol{Y}$ that meets the constraints of ζ, for $p = q$. However, if such a matrix does exist, then it is the smallest possible matrix given these constraints.

4.2.2 Stage 2: Existence of smallest matrix possible.

Now that we have found a way to guarantee that our regular matrix $\boldsymbol{Y}$ is the smallest possible matrix with $girth(\boldsymbol{Y}) = \zeta$, the problem reduces to one of finding a matrix $\boldsymbol{Z}$ with $girth(\boldsymbol{Z}) \geq \zeta$ such that $girth(\boldsymbol{H}) = \zeta$ for the resulting parity check matrix. The matrix $\boldsymbol{Z}$ needs to be regular with column and row weights of $p - 1$.

We use the example of $p = q = 3$ to find the smallest $\boldsymbol{H}$ with girth 4. Fig. 4.3 shows how the matrix Y with $girth(\boldsymbol{Y}) = 4$, formed using the V-construction in Stage 1, is filled with the matrix Z with $girth(\boldsymbol{Z}) = 4$ to obtain matrix $\boldsymbol{H}$ with $girth(\boldsymbol{H}) = 4$. Note that Z is the matrix defined by a single 4-cycle from Fig. 4.1. This is a trivial example that generates an $\boldsymbol{H}$ matrix with all ones. It can easily be generalized that the smallest matrices for any value of $p = q$ and having girth 4 are simply p-by-p matrices with all ones. These matrices are not low density and are full of 4-cycles which makes them effectively useless for BP decoding.

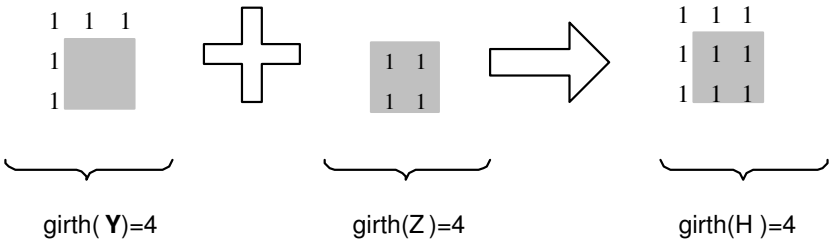


Figure 4.3: Example of Stage 2: Existence of the smallest matrix for $\zeta = 4$ and $p = q = 3$.

Here, we extend the previous example to find the smallest $\boldsymbol{H}$ given girth 6 and $p = q = 3$. First we form a 7-by-7 matrix $\boldsymbol{Y}$ using the V-construction. Now, we just need to find a 4-by-4 matrix $\boldsymbol{Z}$ with $p = q = 2$ and girth at least 6 to fill the remaining part of $\boldsymbol{Y}$. One solution is to use the $\boldsymbol{Z}$ matrix defined by a single 8-cycle from Fig. 4.1.

Fig. 4.4a shows how the matrix $\boldsymbol{Y}$ with $girth(\boldsymbol{Y}) = 6$ is filled with the matrix $\boldsymbol{Z}$ with $girth(\boldsymbol{Z}) = 8$, but this results in a matrix $\boldsymbol{H}$ with $girth(\boldsymbol{H}) = 4$. This shows that the V-construction imposes additional constraints on $\boldsymbol{Z}$ other than requiring $girth(\boldsymbol{Z}) \geq girth(\boldsymbol{Y})$ and a constant column and row weight of $p - 1$. $\boldsymbol{Z}$ must also spread its "1"s such that it does not form cycles of length smaller than

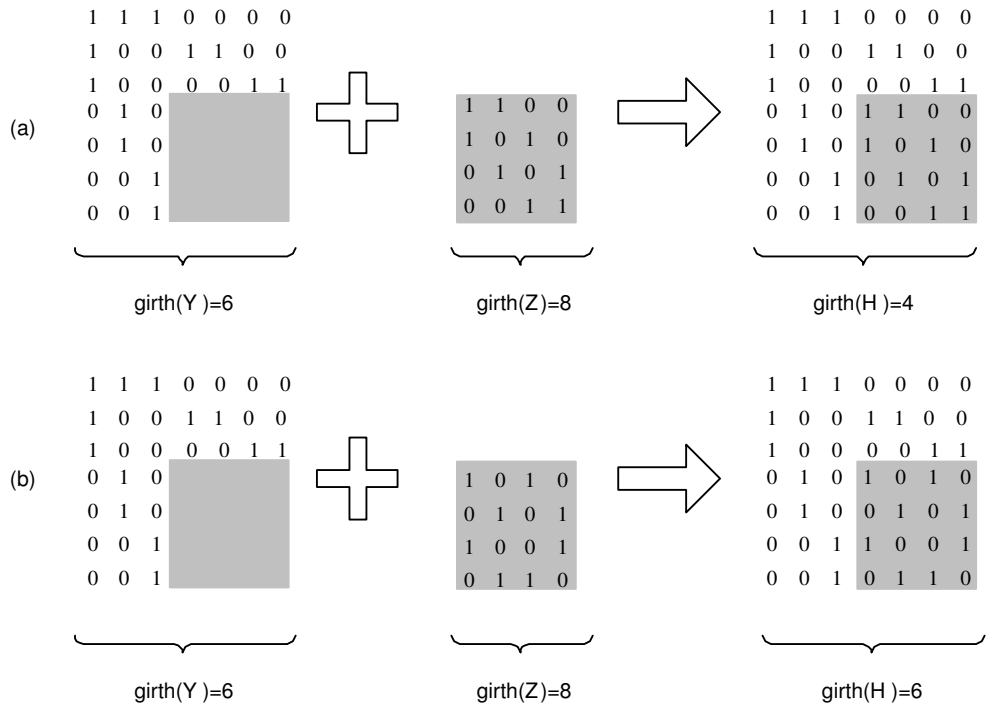


Figure 4.4: Existence of the smallest matrix for $\zeta = 6$ and $p = q = 3$.

girth($\boldsymbol{Y}$), between the "1"s in $\boldsymbol{Z}$ and in $\boldsymbol{Y}$. In other words, for this example, $\boldsymbol{Z}$ must be permutable into a form such that if $\boldsymbol{Z}$ is divided into four 2-by-2 submatrices, each submatrix would contain an even number of "1"s. Each submatrix must also spread its "1"s such that they do not appear in the same row or column within the submatrix. These additional constraints are met by the permutation[5] of $\boldsymbol{Z}$ given in Fig. 4.4b and this generates $\boldsymbol{H}$ with $girth(\boldsymbol{H}) = 6$.

In summary, the V-construction produces the smallest possible regular matrix $\boldsymbol{H}$ for a given girth ζ. However, it does not guarantee the existence of $\boldsymbol{H}$. This depends on the existence of a matrix $\boldsymbol{Z}$ with $girth(\boldsymbol{Z}) \geq \zeta$ and the existence of a suitable permutation of $\boldsymbol{Z}$ to fit into $\boldsymbol{Y}$ such that $girth(\boldsymbol{H}) = \zeta$.

For $\zeta = 6$, the V-construction produces Steiner designs [56, 57]. However, Steiner designs are restricted to constructing girth 6 matrices, whereas the V-construction may be extended to higher girth designs.

[5]If we use $[R1, R2, R3, R4]^T$ and $[C1, C2, C3, C4]$ to label the rows and columns of $\boldsymbol{Z}$ in Fig. 4.4a, respectively, the permutation to get $\boldsymbol{Z}$ in Fig. 4.4b is given by the $[C4, C1, C2, C3]$ sequence column-wise, followed by the $[R3, R2, R4, R1]^T$ sequence row-wise.

4.3 Proposed algorithm for girth 6

Due to the need to find the smallest matrix possible, the V-construction of $\boldsymbol{Y}$ places a very restrictive constraint on the existence of a regular $\boldsymbol{H}$ matrix for a given girth. Here we remove this constraint and focus our effort on finding a regular matrix $\boldsymbol{H}$, without worrying about whether or not the resulting matrix is the smallest possible. We still require the "1"s in $\boldsymbol{H}$ to be spread out evenly, almost similar to the additional constraint of the V-construction on $\boldsymbol{Z}$. This spreading of "1"s in $\boldsymbol{H}$ is reflected in the LDPC code proposed by Gallager in [43], where the $\boldsymbol{H}$ matrix consists of smaller blocks of identity matrices, which have been permuted in a pseudorandom manner, resulting in a girth of 4. Here, we also construct $\boldsymbol{H}$ using smaller blocks of identity matrices. The main difference is that our permutation process is fully systematic (not random). In addition, we can also guarantee that the resulting $\boldsymbol{H}$ will have girth at least 6. The proof is provided in Appendix C.

We now provide a brief outline of our LDPC code construction method, which can create codes with different rates for the same block length. It creates regular LDPC codes, which have parity check matrices, $\boldsymbol{H}$, with a constant row and column[6] weight of p. The construction can be categorized according to the density of "1"s in the generated $\boldsymbol{H}$ matrix. A Density-$\frac{p}{p^q}$ algorithm constructs an $\boldsymbol{H}$ matrix where the density of "1"s is equal to $\frac{p}{p^q}$.

4.3.1 Density-$\frac{p}{p^3}$ Algorithm

In the Density-$\frac{p}{p^3}$ algorithm, we first create a $TableStore$ matrix of size p-by-p^3. This algorithm is described in Fig. 4.5. For $p = 2$, the 2-by-8 $TableStore$ matrix generated is given by

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 & 7 & 8 & 5 & 6 \end{bmatrix}. \tag{4.1}$$

[6]Although setting $p = q$ appears to restrict the (N, K) code parameters where N is the code length and K is the information length, this is not the case. It will be shown in Section 4.4.2 that removing blocks of rows from the parity check matrix, we can set p to be any value smaller than q. This results in different K values for the same N. In Section 8, we point out how removing blocks of columns results in different N values.

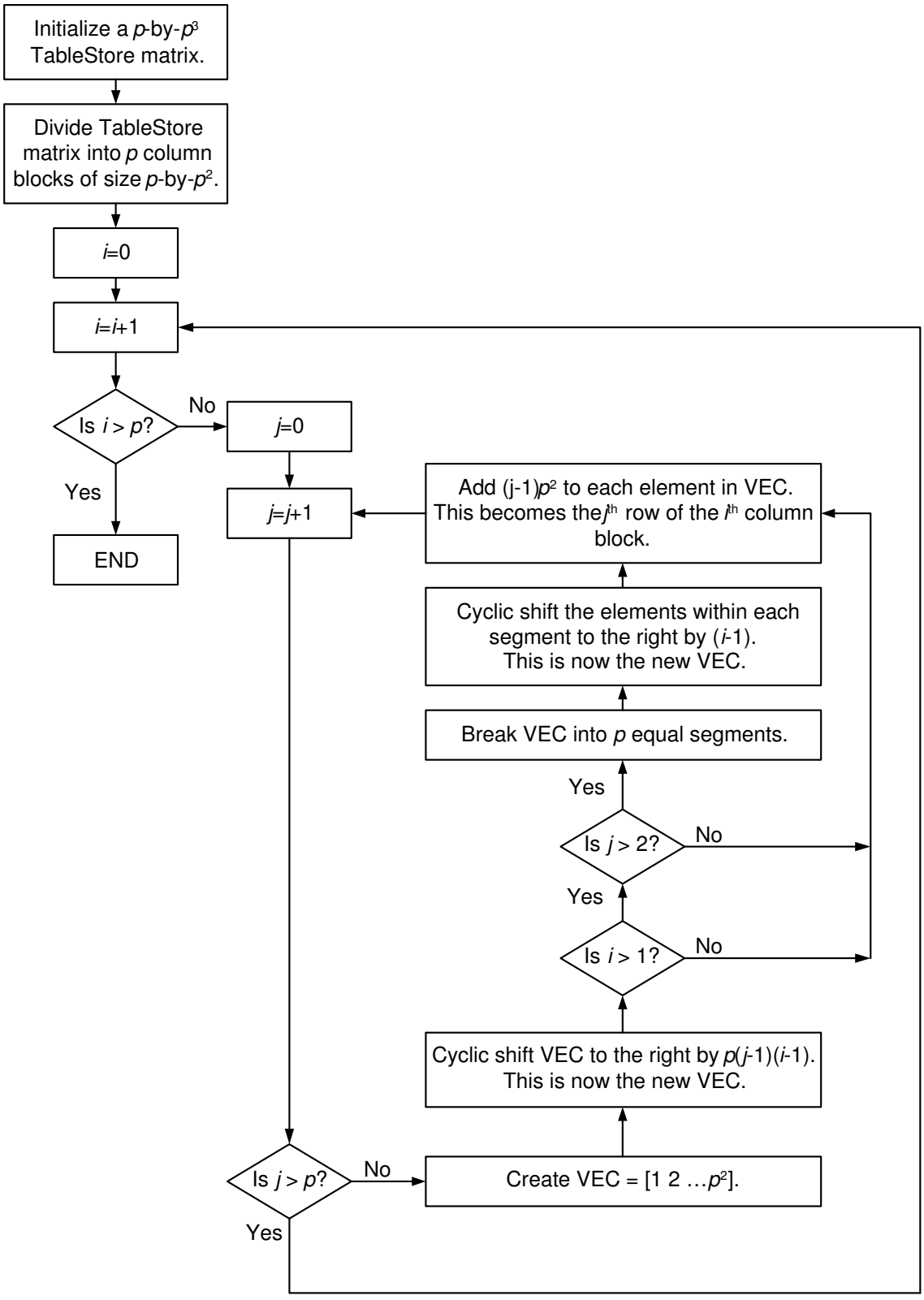


Figure 4.5: Density-$\frac{p}{p^3}$ algorithm to generate the TableStore matrix.

The elements in the k^{th} column of $TableStore$ specify the rows of $\boldsymbol{H}$ containing a "1" in the k^{th} column. The other rows in the same column contain "0"s. The $TableStore$ matrix of equation (4.1) defines the p^3-by-p^3=8-by-8 $\boldsymbol{H}$ matrix,

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{4.2}$$

which has a constant row and column weight of $p = 2$. This construction method can be shown to guarantee an $\boldsymbol{H}$ matrix with girth 6.

4.3.2 Density-$\frac{p}{p^2}$ Algorithm

The Density-$\frac{p}{p^2}$ algorithm is similar to the Density-$\frac{p}{p^3}$ algorithm, but the $\boldsymbol{H}$ matrices constructed are smaller, for the same p. Fig. 4.6 shows the Density-$\frac{p}{p^2}$ algorithm. Here, the $TableStore$ matrix has size p-by-p^2. For $p = 2$, the 2-by-4 $TableStore$ matrix generated is given by

$$\begin{bmatrix} 1 & 2 & 1 & 2 \\ 3 & 4 & 4 & 3 \end{bmatrix}. \tag{4.3}$$

The $TableStore$ matrix of (4.3) defines the p^2-by-p^2=4-by-4 $\boldsymbol{H}$ matrix,

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \tag{4.4}$$

which has a constant row weight and column weight of $p = 2$. This particular construction example generates $\boldsymbol{H}$ with girth 8. In general, the Density-$\frac{p}{p^2}$ algorithm guarantees an $\boldsymbol{H}$ matrix with girth at least 6.

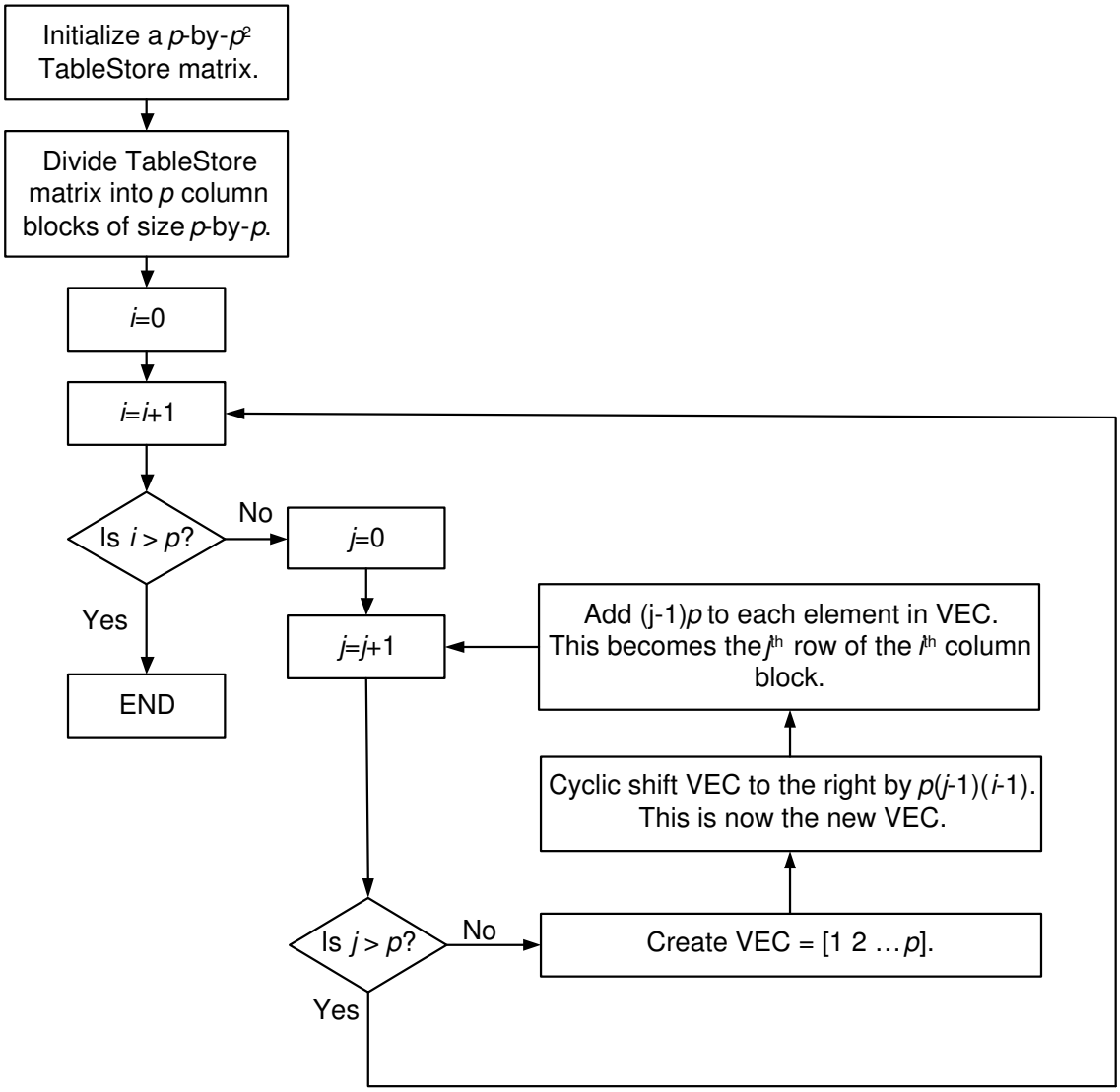


Figure 4.6: Density-$\frac{p}{p^2}$ algorithm to generate the TableStore matrix.

4.4 Code properties

Having constructed the parity check matrix $\boldsymbol{H}$, we now focus on constructing the generator matrix $\boldsymbol{G}$. First we describe two different methods to obtain $\boldsymbol{G}$ from $\boldsymbol{H}$. Then we look at the code properties for the codes constructed using $\boldsymbol{H}$.

4.4.1 Obtaining the Generator Matrix, G

The rows of $\boldsymbol{H}$ in equations (4.2) and (4.4) are not linearly independent. Row reduction is performed on $\boldsymbol{H}$ to obtain a smaller $(N-K)$-by-N matrix $\boldsymbol{H}_{red}$. The rows of $\boldsymbol{H}_{red}$ are linearly independent. $\boldsymbol{H}_{red}$ is used to obtain the generator matrix for the code, $\boldsymbol{G}$, where $\boldsymbol{G}\boldsymbol{H}_{red}^T = \boldsymbol{0}$ and $(.)^T$ denotes a matrix transpose. Note also that $\boldsymbol{G}\boldsymbol{H}^T = \boldsymbol{0}$. Row reduction usually reduces the girth, $girth(\boldsymbol{H}_{red}) \leq girth(\boldsymbol{H})$. Therefore, BP decoding, which does not perform well on LDPC codes with girth 4, is performed on $\boldsymbol{H}$, not $\boldsymbol{H}_{red}$.

The generator matrices obtained using row reduction on $\boldsymbol{H}$ are usually not low-density. A simpler method exists for obtaining generator matrices without performing row reduction. First, we generate an $\boldsymbol{H_{temp}}$ matrix with girth 6 using the Density-$\frac{p}{p^q}$ algorithm. Then, we simply define the generator matrix as $\boldsymbol{G} = [\boldsymbol{I} \quad \boldsymbol{H_{temp}^T}]$ and the parity check matrix as $\boldsymbol{H} = [\boldsymbol{H_{temp}} \quad \boldsymbol{I}]$, where $\boldsymbol{I}$ is a p^q-by-p^q identity matrix. Appending an identity matrix to $\boldsymbol{H_{temp}}$ does not reduce its girth as no edges are removed from its bipartite graph. Therefore, the resulting $\boldsymbol{H}$ and $\boldsymbol{G}$ matrices both have girth 6 and both are low-density. This subclass of LDPC codes is known as the Low Density Generator Matrix (LDGM) codes [47, 58]. To avoid confusion, we use the term "LDPC codes" to indicate LDPC codes generated by applying row reduction on $\boldsymbol{H}$, and "LDGM codes" to indicate LDPC codes generated by appending an identity matrix to $\boldsymbol{H_{temp}}$.

4.4.2 LDPC code properties

Table 4.1 shows the properties of the proposed LDPC codes using the Density-$\frac{p}{p^3}$ algorithm. For prime values of p greater than 2, we can obtain LDPC codes with girth 6 and good minimum Hamming distance, d_{min}. The upper bound for d_{min} here and in the rest of the section is determined by the minimum weight of the rows in $\boldsymbol{G}$.

p	$\boldsymbol{H}$	$\boldsymbol{G}$	R_{ldpc}	upper bound for d_{min}	true d_{min}
2	$\mathbf{8 \times 8}$	$\mathbf{2 \times 8}$	**0.25**	**4**	**4**
3	$\mathbf{27 \times 27}$	$\mathbf{8 \times 27}$	**0.30**	**8**	**8**
4	64×64	21×64	0.33	8	8
5	$\mathbf{125 \times 125}$	$\mathbf{40 \times 125}$	**0.32**	**20**	-
6	216×216	81×216	0.38	8	-
7	$\mathbf{343 \times 343}$	$\mathbf{96 \times 343}$	**0.28**	**28**	-
8	512×512	247×512	0.48	8	-
9	729×729	232×729	0.32	12	-
10	1000×1000	369×1000	0.37	8	-
11	$\mathbf{1331 \times 1331}$	$\mathbf{280 \times 1331}$	**0.21**	**44**	-
12	1728×1728	891×1728	0.52	16	-
13	$\mathbf{2197 \times 2197}$	$\mathbf{408 \times 2197}$	**0.19**	**52**	-

Table 4.1

LDPC code properties for different values of p, using the Density-$\frac{p}{p^3}$ algorithm.

p	$\boldsymbol{H}$	$\boldsymbol{G}$	R_{ldpc}	upper bound for d_{min}	true d_{min}
2	$\mathbf{4 \times 4}$	$\mathbf{1 \times 4}$	**0.25**	**4**	**4**
3	$\mathbf{9 \times 9}$	$\mathbf{2 \times 9}$	**0.22**	**6**	**6**
4	16×16	8×16	0.50	4	4
5	$\mathbf{25 \times 25}$	$\mathbf{4 \times 25}$	**0.16**	**10**	**10**
6	36×36	15×36	0.42	4	4
7	$\mathbf{49 \times 49}$	$\mathbf{6 \times 49}$	**0.12**	**14**	**14**
8	64×64	40×64	0.63	8	-
9	81×81	20×81	0.25	6	6
10	100×100	37×100	0.37	4	-
11	$\mathbf{121 \times 121}$	$\mathbf{10 \times 121}$	**0.08**	**22**	**22**

Table 4.2

LDPC code properties for different values of p, using the Density-$\frac{p}{p^2}$ algorithm.

True d_{min} values, obtained through exhaustive computer search, were obtained only for small values of K (approximately under 30) as the number of searches required is 2^K, where K is the number of rows in the generator matrix.

Table 4.2 shows the properties of the proposed LDPC codes using the Density-$\frac{p}{p^2}$ algorithm. For prime values of p greater than 2, we can obtain LDPC codes with good minimum Hamming distance. This is similar to the trend observed in Table 4.1 where LDPC codes with good minimum Hamming distance are obtained when

p is a prime number. The LDPC codes generated using this algorithm have smaller values of K. Therefore, we are able to use computer searches to find the true d_{min} for $p = 2, 3, 4, 5, 6, 7, 9, 11$. Comparing the true d_{min} values with their upper bounds, we conjecture that our simple upper bound is a good approximation to the true d_{min}.

The upper bound for d_{min} generated by the Density-$\frac{p}{p^3}$ algorithm appears to be approximately $4p$ (for prime values of $p > 5$). This is only $2p$ for the Density-$\frac{p}{p^2}$ algorithm (for all prime values of p observed). However, the corresponding $\boldsymbol{H}$ and $\boldsymbol{G}$ matrices also grows much more quickly with p for the Density-$\frac{p}{p^3}$ algorithm.

The Density-$\frac{p}{p^q}$ construction algorithm generates a p^q-by-p^q square matrix $\boldsymbol{H}$ which can be divided into p^{q-1}-by-p^{q-1} identity matrices, each of which are permuted in such a way that the girth is at least 6. We know that for prime values of p, this algorithm generates LDPC codes with good minimum Hamming distances. However, the corresponding code rates are poor, compared to those obtained with non-prime values of p. This is because the permutation of the p^{q-1}-by-p^{q-1} identity matrices within $\boldsymbol{H}$ results in a large number of independent rows in $\boldsymbol{H}$, or a large rank($\boldsymbol{H}$)=$N - K$. After row reduction, this results in a low value of K, and hence low code rate, R_{ldpc}.

We can increase R_{ldpc} by removing rows from $\boldsymbol{H}$, resulting in a lower value of $(N - K)$ and hence a higher value of K after row reduction[7]. This is shown in Tables 4.3 and 4.4, where blocks[8] of p^{q-1} rows are removed from $\boldsymbol{H}$, beginning with the bottom block and moving upwards.

Tables 4.3 and 4.4 also show that removing rows from $\boldsymbol{H}$ may reduce d_{min}. Therefore, for a fixed N, a tradeoff exists between high R_{ldpc} and high d_{min}. This systematic row removal procedure of starting from the bottom rows and moving upwards was performed to maintain simplicity. It may not be the optimum way of removing rows to generate the best tradeoff between R_{ldpc} and d_{min}, for a given number of rows to be removed. However, it appears to be a good method that yields codes with the desired properties.

[7] K is a non-decreasing value, depending on the number of rows removed and which rows were removed. Removing rows from $\boldsymbol{H}$ does not affect the value of N, which is determined by the number of columns in $\boldsymbol{H}$.

[8] Removing blocks of p^{q-1} rows maintains the regularity of $\boldsymbol{H}$. In general, any number of rows may be removed, but this may result in an irregular $\boldsymbol{H}$.

x	$\boldsymbol{H}$	$\boldsymbol{G}$	R_{ldpc}	upper bound for d_{min}
0	343×343	96×343	0.28	28
49	294×343	126×343	0.37	28
98	245×343	156×343	0.45	28
147	196×343	186×343	0.54	14
196	147×343	216×343	0.63	8
245	98×343	252×343	0.73	4

Table 4.3
LDPC code properties after removing the last x rows of $\boldsymbol{H}$ for $p = 7$, using the Density-$\frac{p}{p^3}$ algorithm.

x	$\boldsymbol{H}$	$\boldsymbol{G}$	R_{ldpc}	upper bound for d_{min}
0	1331×1331	280×1331	0.21	44
121	1210×1331	370×1331	0.28	44
242	1089×1331	460×1331	0.35	44
363	968×1331	550×1331	0.41	44
484	847×1331	640×1331	0.48	44
605	726×1331	730×1331	0.55	44
726	605×1331	820×1331	0.62	24
847	484×1331	910×1331	0.68	14

Table 4.4
LDPC code properties after removing the last x rows of $\boldsymbol{H}$ for $p = 11$, using the Density-$\frac{p}{p^3}$ algorithm.

This ability to generate different code rates for the same block length and in many cases for no loss in minimum distance is important in our proposed BMCM system (c.f. Chapter 5) when using higher order modulation. This will be discussed in more detail in Chapter 6. The Density-$\frac{p}{p^q}$ algorithms can be extended to construct $\boldsymbol{H}$ matrices with girth 8. Larger girths enable LDPC codes to approach its ML performance using a low complexity BP decoding algorithm. However, the LDPC codes generated by the resulting girth-8 $\boldsymbol{H}$ matrices have relatively poor rates and poor minimum Hamming distance.

4.4.3 LDGM code properties

Table 4.5 shows the properties of LDGM codes generated using the Density-$\frac{p}{p^2}$ algorithm. Compared to the LDPC code properties in Table 4.2, it is clear that LDGM codes in general have higher code rates but poorer minimum Hamming distance. This is consistent with the results in [47]. LDGM code properties can be easily generalized. Both the systematic $\boldsymbol{H}$ and $\boldsymbol{G}$ are p^2-by-$2p^2$ matrices with $d_{min} \leq p+1$.

p	$\boldsymbol{H}$	$\boldsymbol{G}$	R_{ldgm}	upper bound for d_{min}
2	4×8	4×8	0.5	3
3	9×18	9×18	0.5	4
4	16×32	16×32	0.5	5
5	25×50	25×50	0.5	6
6	36×72	36×72	0.5	7
7	49×96	49×96	0.5	8
8	64×128	64×128	0.5	9
9	81×162	81×162	0.5	10
10	100×200	100×200	0.5	11
11	121×242	121×242	0.5	12

Table 4.5
LDGM code properties for different values of p generated using the Density-$\frac{p}{p^2}$ algorithm.

4.5 Performance Results in AWGN

The performance of the LDPC codes constructed using the Density-$\frac{p}{p^q}$ algorithms are shown. All simulations presented in this section use BPSK over an AWGN channel. Each simulation point contains 1000 bit errors. The noise variance from equation (A.10) can be simplified to

$$\sigma^2 = \frac{N_0}{2} = \frac{1}{2R_{ldpc}10^{0.1SNR}}, \tag{4.5}$$

since each constellation point has unit energy. Two decoding schemes are used throughout this section. The first is a soft ML scheme, which searches through all possible codewords and selects the codeword at minimum Euclidean distance to the

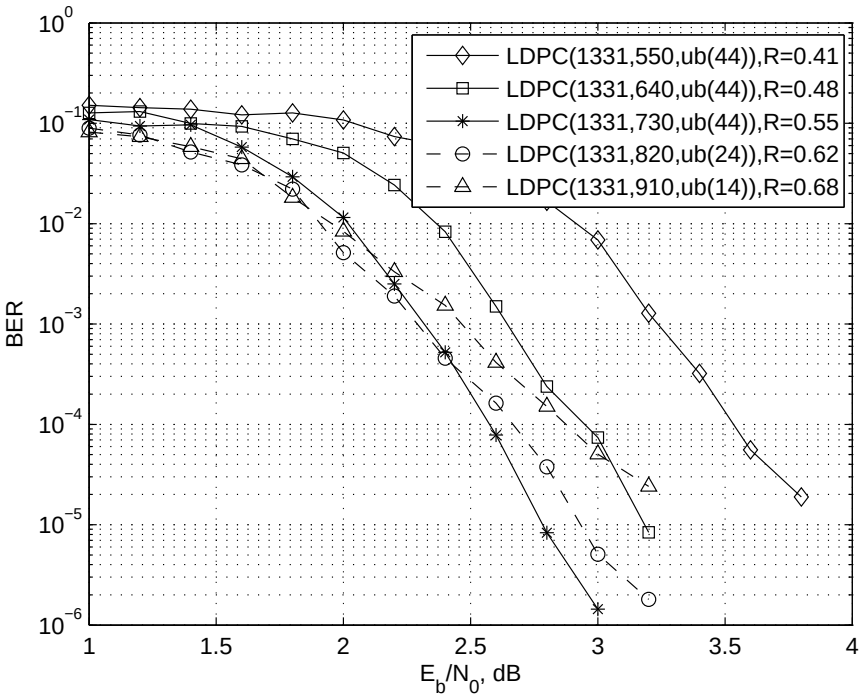


Figure 4.7: BER performance of the proposed LDPC codes for $n = 1331$ and different values of k.

received codeword as the most likely codeword to have been transmitted. This is an optimal scheme, but it has exponentially increasing decoding complexity with K. The second decoding scheme is the BP algorithm, which is suboptimal, but can achieve good performance at much lower complexity. The calculation of the likelihoods for BP decoding is described in [59]. A maximum of 200 BP iterations are performed for each codeword transmitted. The BP algorithm may terminate early if a valid codeword is found.

4.5.1 Performance of the proposed LDPC codes

Fig. 4.7 shows the BER performance of the class of LDPC code generated using the Density-$\frac{p}{p^3}$ algorithm with $p = 11$. Different code rates ranging from 0.41-0.68 are shown, for fixed $N = 1331$. At low SNR, codes with higher rates perform better than those with lower rates, due to the rate penalty in equation (4.5). The channel is so noisy that the better error protection of the lower rate codes is rendered useless. At high SNR, performance is dominated by d_{min}, which increases the slope of the

BER curve. From Fig. 4.7, we can see that codes with a larger upper bound on d_{min} have significantly better slopes than those with a smaller d_{min}. In Table 4.2, we have shown that this upper bound is a good approximation to the true d_{min}. To avoid confusion from here on, the terms "$ub(d_{min})$" and "d_{min}" are used to denote the upper bound on d_{min} and the true d_{min}, respectively.

If no rows were removed from the $\boldsymbol{H}$ generated using the Density-$\frac{p}{p^3}$ algorithm with $p = 11$, we obtain a (1331,280,ub(44)) LDPC code. Each time we remove a multiple of $p^2 = 121$ rows from $\boldsymbol{H}$, starting from the bottom and moving upwards, we increase K by 90. Table 4.4 shows that $ub(d_{min})$ remains at 44 as we increase K from 280 to 730. As K is increased to 820 and 910, $ub(d_{min})$ decreases to 24 and 14, respectively. In Fig. 4.7, we use solid lines for the curves associated with K in the range 280-730, and dashed lines for those associated with K=820 and 910. The solid lines show that we should use the largest K in this range, namely $K = 730$, because we are getting higher rate, higher $ub(d_{min})$ and also better BER performance. We define the threshold rate, R_{th}, as the maximum code rate R_{ldpc} such that $ub(d_{min})$ equals the $ub(d_{min})$ shown in Table 4.1. At BER=10^{-5}, the best performance is given by the (1331,730) LDPC code, which has rate $R_{th} = 0.55$. Above this rate, $ub(d_{min})$ starts to decrease significantly, and a tradeoff appears between R_{ldpc} and $ub(d_{min})$. This is reflected in a slope reduction of the dashed BER curves, due to reduced $ub(d_{min})$, as we increase R_{ldpc}.

Fig. 4.8 shows the BER performance of the set of LDPC codes generated using the Density-$\frac{p}{p^3}$ algorithm with $p = 13$. Row removals are applied to $\boldsymbol{H}$ to obtain a range of rates from 0.55-0.73 for fixed $N = 2197$. Once again, we observe that at high SNR, performance is dominated by the codes with a large minimum distance. As we increase R_{ldpc} beyond R_{th}=0.61, $ub(d_{min})$ starts to decrease significantly. At BER=10^{-5}, the best performance is given by the (2197,1332) LDPC code with rate $R_{th} = 0.61$.

The threshold rate, R_{th}, is an important parameter because it points out the range of R_{ldpc} that we need to consider. LDPC codes with $R_{ldpc} \leq R_{th}$ waste throughput. They do not achieve any performance gain for the reduced code rate. The threshold rates for LDPC codes generated using the Density-$\frac{p}{p^3}$ algorithm with $p = 7, 11, 13$ are $R_{th} = 0.45, 0.55, 0.61$, respectively. Table 4.1 shows that $ub(d_{min})$

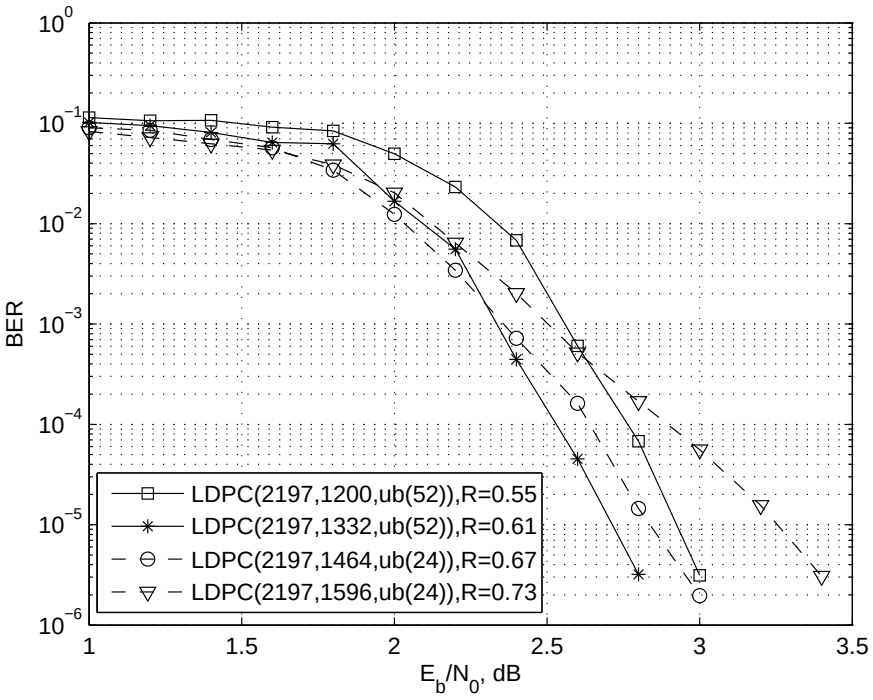


Figure 4.8: BER performance of the proposed LDPC codes for $n = 2197$ and different values of k.

increases with p, for prime values of p. These results suggest that longer LDPC codes will produce better tradeoffs between R_{ldpc} and $ub(d_{min})$, compared to shorter codes.

Fig. 4.9 compares the performance of the (343,186), (1331,730) and (2197,1332) LDPC codes. These are the best codes that were found in terms of BER performance, for their respective block lengths. At BER=10^{-5}, we can see that as the block length increases, the BER performance improves and the code rate increases. This reiterates the advantage of using longer LDPC codes, which is consistent with other types of LDPC codes found in the literature [26, 46].

4.5.2 Performance comparison to other LDPC codes

We now compare the performance of our proposed LDPC codes to other types of LDPC codes found in the literature. We start with the original LDPC code proposed by Gallager in [43]. It was shown in [46] that this code has $N = 20$, $K = 7$ and $d_{min} = 6$.

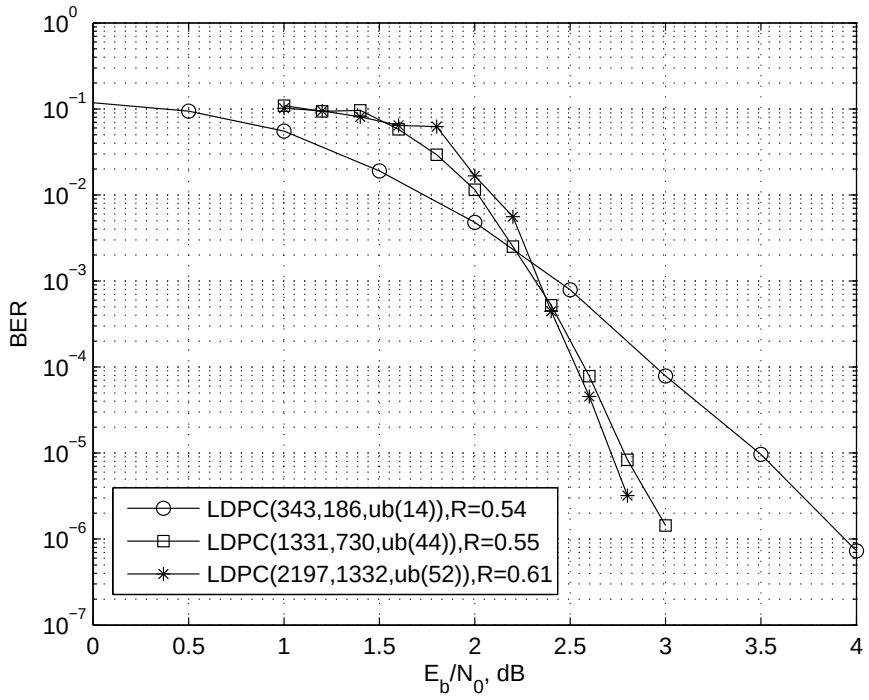


Figure 4.9: BER performance comparison between the best LDPC code for $N = 343, 1331, 2197$.

Fig. 4.10 compares the (20,7,6) Gallager LDPC against our proposed (27,9,6) LDPC[9]. Both LDPC codes were decoded using soft ML decoding and BP decoding with 20 iterations. The BER curves for both LDPC codes have the same slope at high SNR because they have the same d_{min}. However, at BER=10^{-4}, our LDPC performs about 0.3dB better when soft ML decoding is performed. We should note that our LDPC code has a slightly longer length and a slightly lower code rate than the Gallager LDPC. When BP decoding is performed, our LDPC code converges about 0.1dB away from its ML performance and the Gallager LDPC code converges about 0.5dB away. This is because the Gallager $\boldsymbol{H}$ matrix has girth 4, whereas our $\boldsymbol{H}$ matrix has girth 6.

Sparse circulant matrices may be used to construct a sparse $\boldsymbol{H}$ matrix [49, 46]. In [49], a (510,256) LDPC code, C_0, was constructed using sparse circulant matrices. This code contains many 4-cycles. A trellis-based method for removing short cycles was introduced in [49]. This is an exhaustive method that identifies all the short

[9]This is the fairest LDPC we could find in terms of similar (N,K,d_{min}) properties.

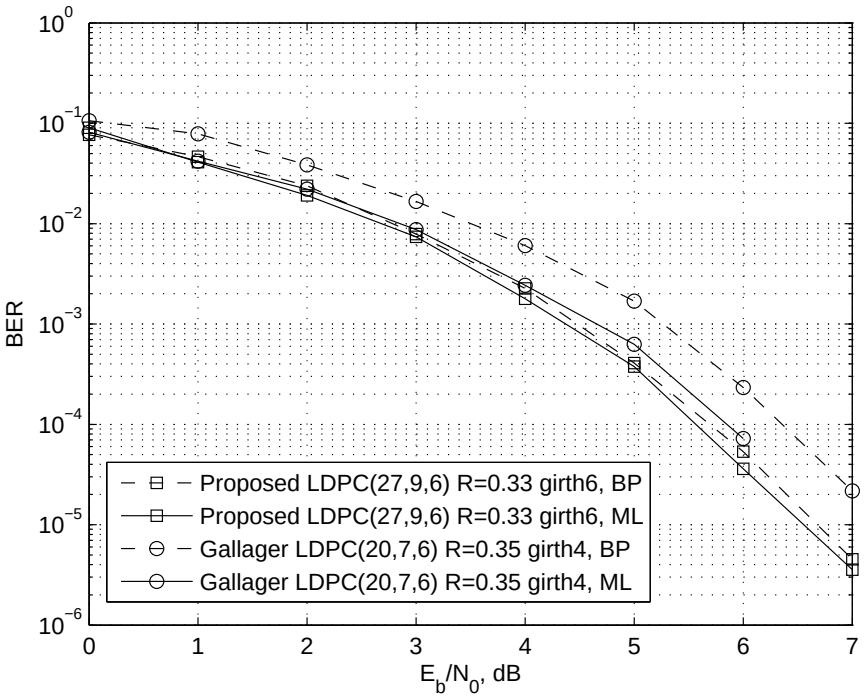


Figure 4.10: BER performance of the proposed (27,9,6) LDPC code and the (20,7,6) LDPC code by Gallager. Soft brute force (ML) decoding and BP decoding were used on each code.

cycles and removes them by performing column/row splitting on $\boldsymbol{H}$. Applying this cycle-removal method on C_0 produces a (510,255) LDPC code, C_1, with girth 6. Repeating this process on C_1 produces a (510,255) LDPC code, C_2, with girth 8. Fig. 4.11 shows the performance of the C_0, C_1 and C_2 LDPC codes and our proposed (343,186) LDPC code. These four codes achieve BER=10^{-5} at 4.8dB, 4.3dB, 3.2dB and $3.5dB$, respectively. It is obvious from the slopes of the BER curves that our LDPC code has a smaller d_{min} compared to the other codes. However, note that our LDPC code has a higher code rate (0.54 versus 0.50) and a shorter block length (343 versus 510), compared to the C_0, C_1 and C_2 LDPC codes of [49].

Recently, LDPC codes have been constructed based on finite geometries. Compared to Gallager's pseudorandom LDPC, these codes have a very structured algebraic construction and have been shown to have relatively good minimum distances and girth 6 [52, 46]. One class of the finite-geometry LDPC codes is the Euclidean Geometry LDPC (EG-LDPC) codes. It was shown in [46] that we can obtain an even better class of codes, called the extended EG-LDPC codes by ap-

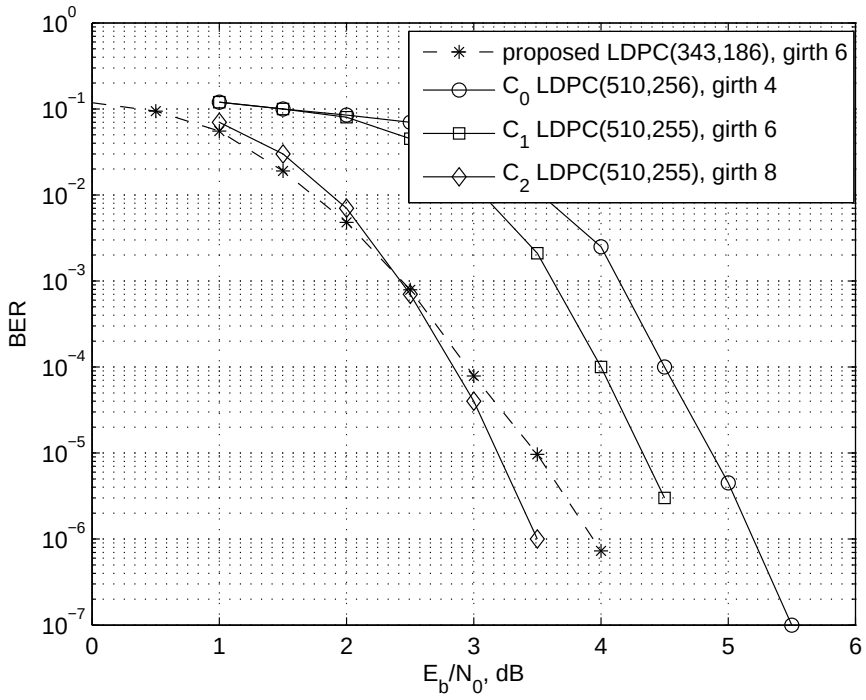


Figure 4.11: BER performance of the proposed (343,186) LDPC code, the (510,256) C_0, (510,256) C_1 and (510,256) C_2 LDPC codes from [49].

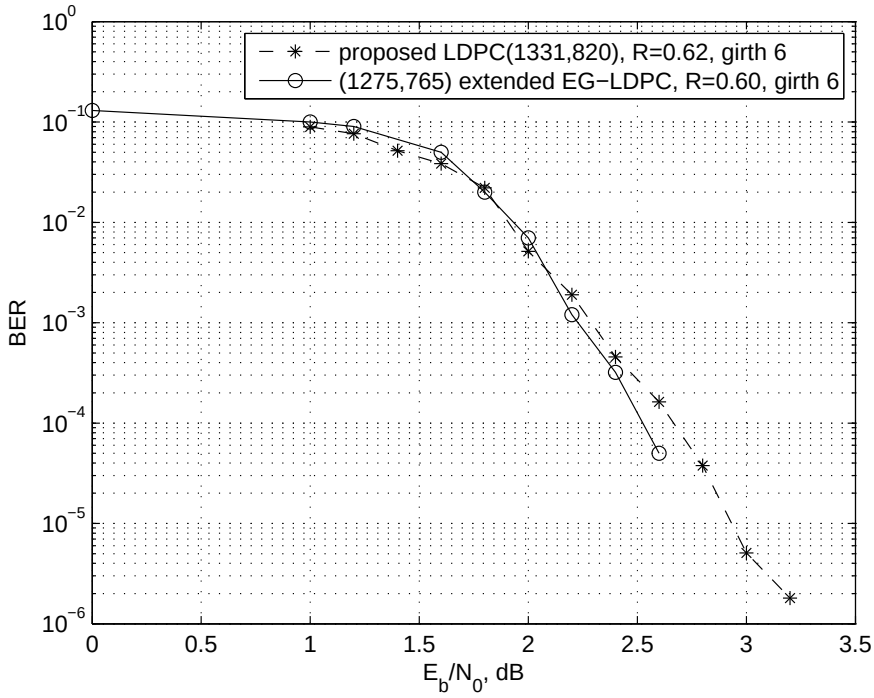


Figure 4.12: BER performance of the proposed (1331,820) LDPC code and the (1275,765) extended EG-LDPC code from [49].

plying row and column splitting to the $\boldsymbol{H}$ matrix of a standard EG-LDPC code. In Fig. 4.12, we show the performance of a (1275,765) extended EG-LDPC code and our proposed (1331,820) LDPC code, which does not require any row and column splitting. Our proposed LDPC performs approximately 0.2dB worse compared to the extended EG-LDPC code at a BER of 10^{-4}. However, our LDPC code has a significantly simpler construction.

4.6 Special Cases

4.6.1 Dual Hamming Codes

The $\boldsymbol{H}$ matrix in Fig. 4.4b has girth 6. Row reduction reduces this matrix to a 4-by-7 $\boldsymbol{H_{red}}$ matrix, as shown in Fig. 4.13, and this corresponds to the (7,3,4) dual Hamming Code. The (7,3,4) dual Hamming code satisfies the necessary condition[10] [60] for the existence of a 4-cycle free Tanner graph.

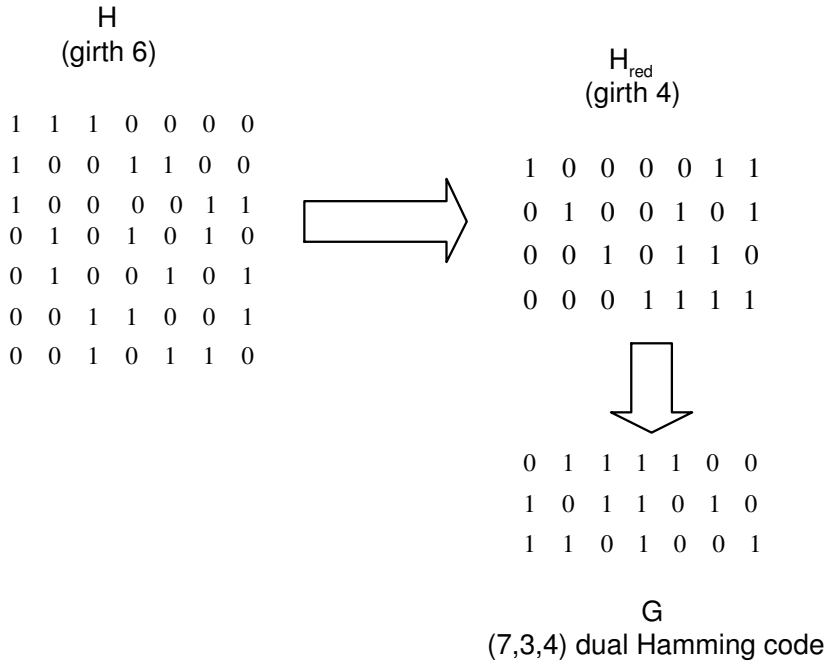


Figure 4.13: The relationship between $\boldsymbol{H}$ with girth 6, $\boldsymbol{H_{red}}$ with girth 4, and $\boldsymbol{G}$ for the (7,3,4) dual Hamming code.

[10]This necessary condition is only mentioned here because it requires the knowledge of the minimum distance of the dual code.

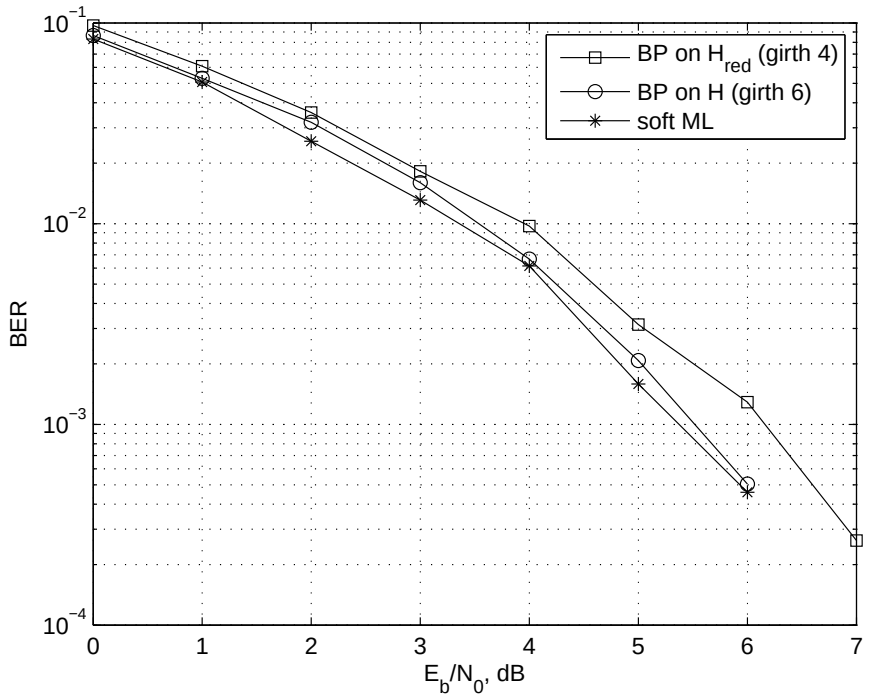


Figure 4.14: BER performance of the (7,3,4) dual Hamming code with different decoding schemes, using BPSK in AWGN.

If each row of $\boldsymbol{H}$ represent a node, and each column of $\boldsymbol{H}$ represent a line, and each "1" in $\boldsymbol{H}$ denotes the nodes that each line passes through, then $\boldsymbol{H}$ can be represented as a Fano Diagram [46]. The rows of $\boldsymbol{H}$ are codewords of the (7,4,3) Hamming code, and their relationship to the Fano Diagram was shown in [46].

Fig. 4.14 shows the BER performance of the (7,3,4) dual Hamming code using soft brute force decoding which achieves ML, and BP decoding on $\boldsymbol{H}$ and $\boldsymbol{H_{red}}$. At BER=10^{-3}, the performance of BP decoding using $\boldsymbol{H}$ is 0.1dB away from ML performance, whereas that using the row reduced $\boldsymbol{H_{red}}$ is 0.8dB away from ML. Row reduction decreases the girth of $\boldsymbol{H}$ from 6 to 4. The presence of 4-cycles in $\boldsymbol{H_{red}}$ causes BP decoding to converge away from ML performance. The degrading effect of 4-cycles was also observed in [49]. This illustrates why we decode using $\boldsymbol{H}$ even though $\boldsymbol{H_{red}}$ is used to create $\boldsymbol{G}$.

4.6.2 Single Parity Check Codes

It was shown in Fig. 4.3 that the smallest matrix with column and row weight of 3 and girth 4 is a 3-by-3 matrix of all ones

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{4.6}$$

Row reduction generates a systematic parity check matrix

$$\boldsymbol{H_{red}} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{4.7}$$

and a systematic generator matrix

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{4.8}$$

This is the generator matrix for a (3,2,2) single-parity check (SPC) code.

In general, the smallest matrix for any $p = q$ value and having girth 4 is a p-by-p matrix of all ones. Row reduction generates a 1-by-p parity check matrix $\boldsymbol{H_{red}}$ and a $(p-1)$-by-p generator matrix $\boldsymbol{G}$. This defines all the (p,$p-1$,2) SPC codes.

4.6.3 2-Dimensional Product SPC Codes

The LDGM codes generated using $\boldsymbol{H_{temp}}$ from the Density-$\frac{p}{p^2}$ algorithm in Fig. 4.6 has girth 6 and $R_{ldpc} = \frac{1}{2}$. For $p = 2$, the Density-$\frac{p}{p^2}$ algorithm generates

$$\boldsymbol{H_{temp}} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \tag{4.9}$$

which happens to have girth 8. This is also the definition of an 8-cycle, as shown in Fig. 4.1. Appending an identity matrix to $\boldsymbol{H_{temp}}$ results in a systematic parity check matrix

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

and a systematic generator matrix

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{4.11}$$

Multi-dimensional product codes increase the minimum distance of the component codes but reduce the code rate [61, 62, 63]. Fig. 4.15a shows the 2-Dimensional product SPC (2D-Product-SPC) codes, without the parity-on-parity bit [62]. All references to 2D-Product SPC codes in this thesis assume that no parity-on-parity bit is used. Four information bits $u1, u2, u3, u4$ are encoded into parity bits $p5, p6$ in one dimension and $p7, p8$ in the other dimension. The encoding is done using SPC codes, and the corresponding SPC equations are shown in Fig. 4.15b. These equations can be translated into a systematic generator matrix, $\boldsymbol{G}$, as shown in Fig.4.15c. This $\boldsymbol{G}$ is the same as the generator matrix constructed using the Density-$\frac{p}{p^2}$ algorithm for $p = 2$, as shown in equation (4.11).

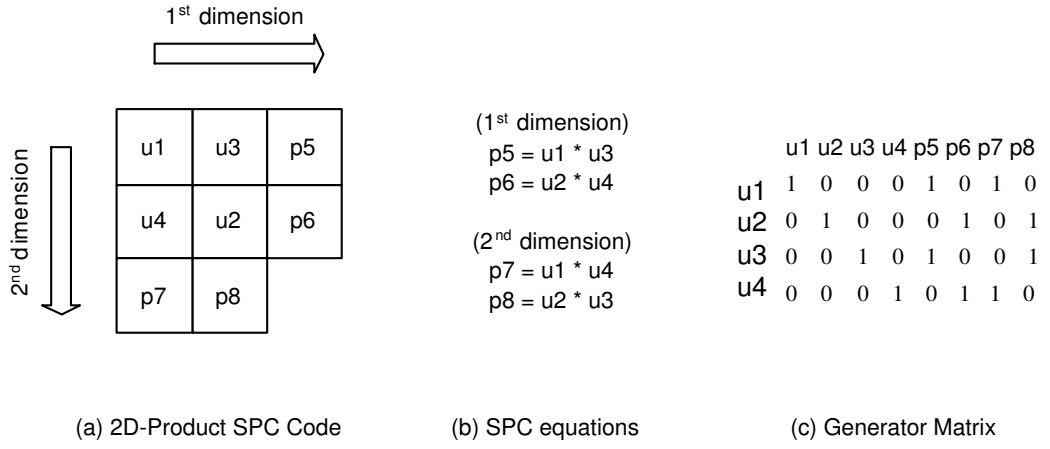


Figure 4.15: The relationship between the SPC equations and the corresponding generator matrix for a 2D-product SPC code, without the parity-on-parity bit. The * symbol denotes an exclusive-OR operation.

This similarity to a 2D-product SPC code can be generalized to a class of LDGM codes using the following procedure: (1) First generate $\boldsymbol{H_{temp}}$ using the Density-$\frac{p}{p^2}$ algorithm. (2) Remove the last $(p-2)p$ rows of $\boldsymbol{H_{temp}}$ to form a regular $2p$-by-p^2 matrix $\boldsymbol{H_{temp2}}$ with column weight 2 and row weight p. (3)

Form the systematic parity check and generator matrices for the LDGM code as $\boldsymbol{H} = [\boldsymbol{H}_{temp2} \quad \boldsymbol{I}_{2p}]$ and $\boldsymbol{H} = [\boldsymbol{I}_{p^2} \quad \boldsymbol{H}_{temp2}^T]$, respectively, where $\boldsymbol{I}_p$ is a p-by-p identity matrix. This procedure generates a class of $(p^2 + 2p, p^2, 3)$ LDGM codes which are equivalent to 2D-product SPC codes, with $R_{ldgm} = \frac{p}{p+2}$.

4.7 Summary

We have developed a simple LDPC construction, which to the best of our knowledge is novel. This construction allows us to generate LDPC codes with a wide range of code lengths N and code rates R_{ldpc}. For fixed N, this construction also allows us to tradeoff R_{ldpc} against d_{min}, which is useful for systems which require a given error rate and system throughput. As will be discussed in Chapters 5 and 6, this is especially useful when high order modulations are used, e.g. 16-QAM and 64-QAM, due to the unequal error protection on each constellation label.

Another advantage of this construction is that the resulting LDPC codes have girth 6. Row removals reduce the size of $\boldsymbol{H}$, which may increase the girth even further. The absence of 4-cycles allows low-complexity BP decoding to be used to get near-ML performance. The LDPC codes generated here will be used as the component codes for our proposed BMCM-STBC structure in Chapter 5. Longer LDPC codes also provide a better tradeoff between R_{ldpc} and $up(d_{min})$.

Compared to other types of LDPC codes, our proposed construction is relatively simple. Despite the simple construction, it was shown that the proposed LDPC codes perform reasonably well, compared to other types of LDPC code.

Chapter 5

High Throughput BMCM-QOSTBC-PIC Systems

5.1 Motivation

Direct transmission of coded modulation schemes [20, 21, 22, 23] in multiple-input multiple-output (MIMO) systems achieves high rates, at the expense of high detection complexity. The concatenation of coded modulation schemes with orthogonal space-time block codes (OSTBCs) as in [64, 65] separates the transmitted symbols at the receiver, thus reducing the detection complexity. However, OSTBCs cannot offer rates greater than 1. To increase throughput, we consider quasi-orthogonal space-time block codes (QOSTBCs) [17, 16], which can provide higher rates than OSTBCs. We concatenate bit-mapped coded modulation (BMCM) with the QOSTBCs. This BMCM-STBC structure uses parallel forward error correction (FEC) component codes, which allows parallel decoding in the receiver. Unlike bit-interleaved coded modulation (BICM) [35], BMCM does not require interleaving and so almost always has shorter processing delays[1]. We use parallel low density parity-check (LDPC) component codes in the BMCM structure. This allows belief propagation (BP) decoding, which has modest decoding complexity, to be performed in parallel for each component code. We employ the LDPC code construction of Chapter 4 as it has the ability to generate many different code rates for a fixed block length. This enables a wide range of throughput versus performance tradeoffs.

[1] A direct comparison between BICM and BMCM is provided in Appendix B.

We focus on the double space time transmit diversity (DSTTD) scheme of [16]. Essentially it is a Rate 2 QOSTBC that combines the Alamouti OSTBC [8] with the Bell-Labs Layered Space-Time (BLAST) Architecture of [5]. Similar hybrid space-time coding schemes can be found in [7, 66]. The BLAST aspect of these schemes introduces co-channel interference which complicates detection, and as a result ordered successive interference cancellation (SIC) is usually employed [16, 66, 7].

Here, we employ iterative parallel interference cancellation (PIC) [28, 67, 68] to mitigate the effects of co-channel interference. PIC is widely used in multi-user environments and is considered less complex than SIC [28]. As in [67, 68], iterations will be performed between the PIC detector and parallel FEC decoders. Diagonal-BLAST and convolutional codes are used in [68]. A performance comparison of layered space-time (LST) codes is carried out in [67], where an improved version of the PIC detector with decision statistics combining [69] is used. To maintain simplicity, we use a standard PIC detector which is sufficient to obtain good performance.

Section 5.2 describes the proposed BMCM-STBC structure using the LDPC codes from Chapter 4. Three different MIMO channel models are presented in Section 5.3. In Section 5.4, two different detection-decoding schemes are presented. The first employs an optimal one-off joint detection (JD) scheme [17, 25] followed by BP decoding. The second uses a joint iterative PIC-BP [67, 68] decoding scheme which is suboptimal, but has much lower complexity. In Section 5.5, we develop a simplified calculation of bit metrics for BP decoding. Section 5.6 describes the different ways we can terminate the iterative PIC-BP decoding scheme.

The main contribution of this chapter is the use of the PIC scheme with QOSTBCs. The PIC scheme is used in an iterative structure together with BP decoding. In [67, 68], the PIC scheme is only applied to LST codes. In Chapter 6, the quasi-orthogonality of the Rate 1 QOSTBC in equation (2.5) and Rate 2 QOSTBC in equation (2.6) is shown to provide a performance gain over the non-orthogonal LST codes. This is due to fewer co-channel interferers in the QOSTBCs.

5.2 Proposed BMCM-QOSTBC Structure

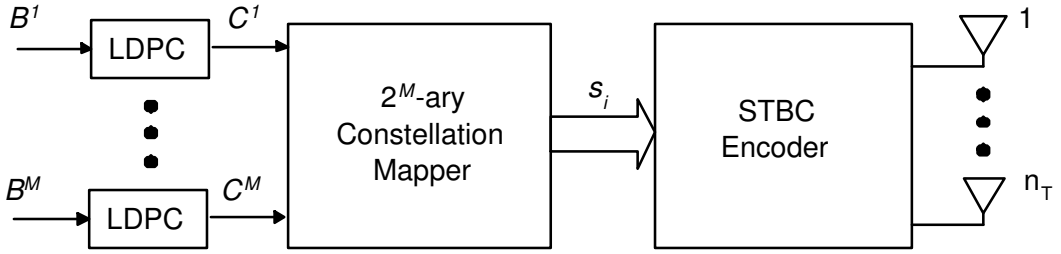


Figure 5.1: BMCM-STBC encoder structure.

The proposed BMCM-STBC encoder structure is shown in Fig. 5.1 for any 2^M-ary constellation. The input data stream is demultiplexed into M data substreams containing data blocks $\{\boldsymbol{B}^m\}_{m=1}^M$. The m^{th} data substream has block length K_m. Each substream is encoded using a forward error-correcting (FEC) code to obtain a set of M length-N codewords $\{\boldsymbol{C}^m\}_{m=1}^M$, where the i^{th} encoded bit in $\boldsymbol{C}^m$ is denoted C_i^m. In the BMCM structure, each substream can be encoded using FEC codes with different values of K_m as long as all M codes produce codewords of the same block length, N. The LDPC code construction from Chapter 4 is a natural choice as it provides the flexibility of tailoring different code rates to different component codes. The overall rate of the M LDPC codes is then

$$R_{ldpc} = \frac{\sum_{m=1}^{M} K_m}{MN}. \tag{5.1}$$

The encoding process can be written as $\boldsymbol{C}^m = \boldsymbol{B}^m\boldsymbol{G}^m$, where $\boldsymbol{G}^m$ is the K_m-by-N generator matrix of the m^{th} LDPC component code. The i^{th} bits from all M codewords collectively select the i^{th} 2^M-ary constellation point s_i. The stream of 2^M-ary constellation points s_i are then "encoded" using a STBC for the required number of transmit antennas, n_T. Here, we focus on systems with $n_T = 4$ and we use the Rate 1 and Rate 2 QOSTBCs described in Chapter 2. However, we can easily extend this system to achieve higher throughput by employing QOSTBCs with[2] $R_{stbc} \geq 3$, at the expense of higher system complexity.

[2]For example, by using 4 Alamouti OSTBCs in parallel, we can transmit 8 new symbols in 2 time slots, giving $R_{stbc} = 4$.

5.3 Channel Models

We consider Rayleigh flat fading or non-frequency selective channel models. Let $\alpha_p^q(t)$ denote the complex fading coefficient affecting the symbol $x_p(t)$ at time t for the subchannel between the p^{th} transmit antenna and the q^{th} receive antenna for $p = 1, 2, ..., n_T$ and $q = 1, 2, ..., n_R$. We model the fading coefficients as zero-mean complex Gaussian random variables with variance $1/2$ per dimension. We assume independent subchannels. We consider three types of time-varying channels (from less to more practical):

Independent quasi-static (QS-Ind) fading channel: This is the most common channel model used in the literature [8, 24, 7, 17], where quasi-static fading is usually assumed. The fading coefficients remain fixed during each STBC block of L time slots and vary independently from one block to the next.

Time-varying quasi-static (QS-fDT) fading channel: Time-varying quasi-static (QS-fDT) fading channel: The fading coefficients remain fixed during each STBC block of L time slots, but vary from block to block according to the normalized fade rate f_DT, where the maximum Doppler shift f_D is normalized to the symbol period T. Using a third-order FIR filter [70], we continuously generate a set of fading coefficients which vary according to f_DT, but we only use every L^{th} coefficient.

Time-varying (NoQS-fDT) fading channel: Time-varying (NoQS-fDT) fading channel: Here, the quasi-static fading assumption is removed and the fading coefficients vary continuously from one time slot to the next according to the normalized fade rate f_DT.

CSI is assumed to be available at the receiver, but not at the transmitter. We fix the total transmitted energy across all n_T transmit antennas to be 1 for each time slot. In the case of $n_T = 4$, the symbol transmitted from each antenna then contains $\frac{1}{n_T} = 1/4$ unit of energy. The sampled signal at each receive antenna is a noisy superposition of the transmitted signals after undergoing flat Rayleigh fading, and is given at sample time t by

$$r^q(t) = \sum_{p=1}^{n_T} \alpha_p^q(t) x_p(t) + w^q(t), \qquad t = 1, ..., L, \tag{5.2}$$

where $w^q(t)$ is the additive white Gaussian noise (AWGN) at the q^{th} receive antenna for $q = 1, 2, ..., n_R$. It is modeled as an independent complex Gaussian random variable with zero mean and a one-dimensional noise variance[3] defined by

$$\sigma^2 = \frac{N_0}{2} = \frac{n_{T,used} n_R E_{s,Tx}}{2MR_{stbc}R_{ldpc}10^{0.1SNR_{total}}}, \tag{5.3}$$

where $E_{s,Tx}$ is the average energy of a constellation symbol from each transmit antenna, $n_{T,used}$ is the number of antennas used for transmission in any time slot and SNR_{total} is the signal-to-noise ratio at the output of the linear processing block (c.f. Figs. 5.2 and 5.3) in decibels (dB). If the SNR per receive antenna is used, the noise variance is defined as $\frac{\sigma^2}{n_R}$.

5.4 Detection-Decoding schemes

Assuming perfect CSI, linear processing produces estimates of the transmitted symbols s_i, given by

$$\begin{aligned}
\hat{s}_1 &= \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_1 + \sum_{q=1}^{n_R} 2(\alpha_1^q\alpha_4^{q*} - \alpha_2^q\alpha_3^{q*})s_4 + noise \\
\hat{s}_2 &= \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_2 + \sum_{q=1}^{n_R} 2(\alpha_2^q\alpha_3^{q*} - \alpha_1^q\alpha_4^{q*})s_3 + noise \\
\hat{s}_3 &= \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_3 + \sum_{q=1}^{n_R} 2(\alpha_2^q\alpha_3^{q*} - \alpha_1^q\alpha_4^{q*})s_2 + noise \\
\hat{s}_4 &= \underbrace{\sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2 + |\alpha_3^q|^2 + |\alpha_4^q|^2)s_4}_{\text{desired terms}} + \underbrace{\sum_{q=1}^{n_R} 2(\alpha_1^q\alpha_4^{q*} - \alpha_2^q\alpha_3^{q*})s_1}_{\text{interference terms}} + noise.
\end{aligned} \tag{5.4}$$

[3]The derivation of this noise variance is provided in Appendix A.

for the Rate 1 QOSTBC and

$$\begin{aligned}
\hat{s}_1 &= \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2)s_1 + \sum_{q=1}^{n_R}(\alpha_1^{q*}\alpha_3^q + \alpha_2^q\alpha_4^{q*})s_3 + \sum_{q=1}^{n_R}(\alpha_1^{q*}\alpha_4^q - \alpha_2^q\alpha_3^{q*})s_4 + noise \\
\hat{s}_2 &= \sum_{q=1}^{n_R}(|\alpha_1^q|^2 + |\alpha_2^q|^2)s_2 + \sum_{q=1}^{n_R}(\alpha_2^{q*}\alpha_4^q + \alpha_1^q\alpha_3^{q*})s_4 + \sum_{q=1}^{n_R}(\alpha_2^{q*}\alpha_3^q - \alpha_1^q\alpha_4^{q*})s_3 + noise \\
\hat{s}_3 &= \sum_{q=1}^{n_R}(|\alpha_3^q|^2 + |\alpha_4^q|^2)s_3 + \sum_{q=1}^{n_R}(\alpha_3^{q*}\alpha_1^q + \alpha_4^q\alpha_2^{q*})s_1 + \sum_{q=1}^{n_R}(\alpha_3^{q*}\alpha_2^q - \alpha_4^q\alpha_1^{q*})s_2 + noise \\
\hat{s}_4 &= \underbrace{\sum_{q=1}^{n_R}(|\alpha_3^q|^2 + |\alpha_4^q|^2)s_4}_{\text{desired terms}} + \underbrace{\sum_{q=1}^{n_R}(\alpha_4^{q*}\alpha_2^q + \alpha_3^q\alpha_1^{q*})s_2 + \sum_{q=1}^{n_R}(\alpha_4^{q*}\alpha_1^q - \alpha_3^q\alpha_2^{q*})s_1}_{\text{interference terms}} + noise.
\end{aligned} \tag{5.5}$$

for the Rate 2 QOSTBC. Since QOSTBCs are not fully orthogonal, these symbol estimates contain co-channel interference. The optimum way to minimize the effects of this interference is to employ JD [17]. Here, we propose the use of an iterative PIC scheme [28]. Although iterative PIC is suboptimal compared to JD, it is shown in Chapter 6 to produce very good performance when FEC codes are employed. As discussed in Chapter 2, PIC has a detection complexity which increases linearly with throughput, compared to an exponential increase for JD.

5.4.1 One-Off JD-BP Decoding

The receiver structure for the JD scheme of [17] is shown in Fig. 5.2. The Rate 1 QOSTBC of [17] and the Rate 2 QOSTBC of [16] produce estimates that contains co-cochannel interference, as shown in equations (5.4) and (5.5) respectively. The JD scheme[4] considers all possible combinations of constellation points consisting of the hypothesized transmitted symbol and the co-channel interferers, and selects the best combination based on Euclidean distance across all symbols. The improved symbol estimates are then demapped into soft bit metrics for the M BP decoders. The derivation of these bit metrics is shown in Section 5.5.

As shown in equation (5.4), linear processing on the Rate 1 QOSTBC decouples the transmitted symbols into two independent pairs. This is a special case that allows a reduced search complexity using pairwise JD. For a BMCM system

[4]A detail description of the JD scheme is given in Section 2.2.2.

using 16-QAM and the Rate 1 QOSTBC, we need to search through $2 \times 16^2 = 512$ possible pairs of constellation points. For the Rate 2 QOSTBC, we need to find the combination of constellation points which minimizes the Euclidean distance, for all four transmitted symbols. Therefore, a BMCM system using 16-QAM and the Rate 2 QOSTBC requires a search through $16^4 = 65536$ possible combinations of four constellation points. Thus we see that the complexity of JD increases as $2(2^M)^2$ for the Rate 1 QOSTBC and $(2^M)^4$ for the Rate 2 QOSTBC.

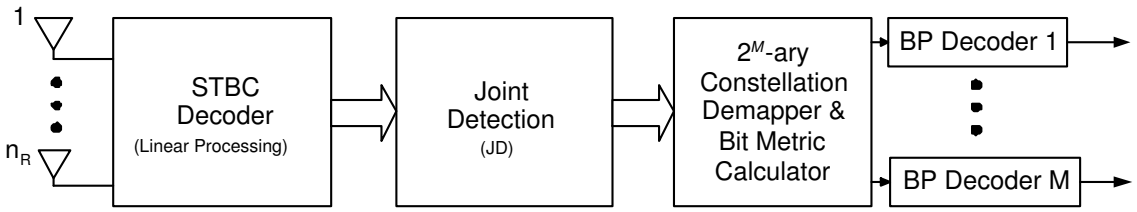


Figure 5.2: Receiver structure using joint detection.

5.4.2 Iterative PIC-BP Decoding

The exponentially increasing complexity of the JD process as I, the number of co-channel interferers, increases motivates the use of a simpler detection scheme, namely PIC. In [68], information is iteratively shared between a PIC block and n_T convolutional decoders, via interleaving/de-interleaving. In [67], information is iteratively shared between the PIC block and n_T LDPC decoders, again via interleaving/de-interleaving. In the present instance, information is iteratively exchanged between the PIC block and the M parallel LDPC decoders, via mapping/demapping. The resulting receiver structure is shown in Fig. 5.3.

In the first iteration, no PIC is performed following linear processing. The symbol estimates from the linear processing block are demodulated and the soft bit metrics (discussed in Section 5.5) are passed directly to the M parallel LDPC decoders. The bit decisions from the LDPC decoders are re-modulated to provide improved estimates of the transmitted symbols, which are then used with the CSI to cancel the interference according to equations (5.4) or (5.5). This PIC update process generates improved symbol estimates which are demodulated and the resulting soft

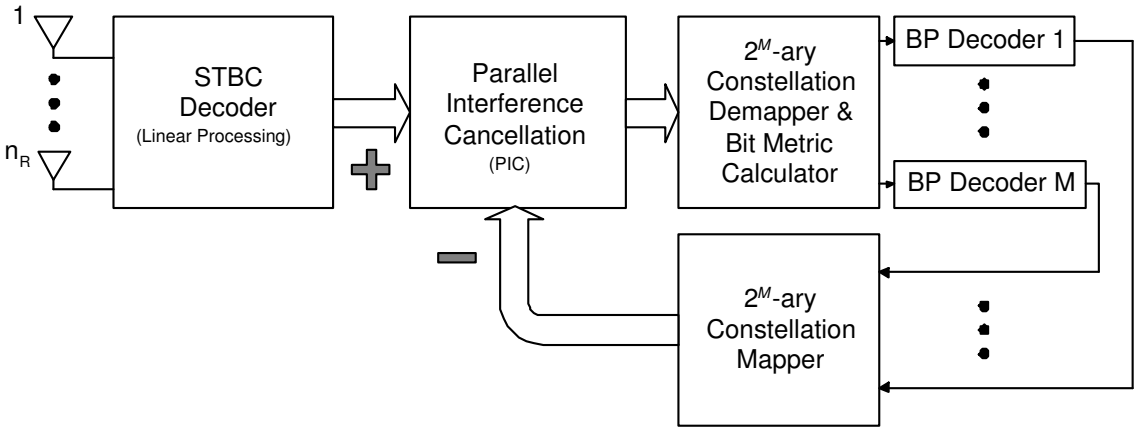


Figure 5.3: Receiver structure using parallel interference cancellation.

bit metrics are passed to the parallel LDPC decoders. This iterative process is repeated until there is negligible further improvement in performance.

The LDPC decoders use the BP decoding algorithm, which requires internal iterations. This is different from the iterations between the PIC block and the LDPC decoders described above. We use the terms BP iterations and PIC updates, respectively, to distinguish between the two iterative processes.

Unlike the SIC schemes of BLAST, no ordering is needed in PIC schemes as interference cancellation is done in parallel, which reduces delay. The number of subtraction operations in each PIC update increases linearly with the number of interferers, I. For example, equation (5.4) shows that the Rate 1 QOSTBC requires four subtraction operations to cancel out the interference in each PIC update. Therefore, if 5 PIC iterations were used, $(I = 1)(4)(5iter) = 20$ subtraction operations are needed. For the Rate 2 QOSTBC, $(I = 2)(4)(5) = 40$ subtraction operations are required. No subtraction operations are needed for JD.

On the other hand, the detection complexity of PIC is only $4(2^M)$ per PIC update for both the Rate 1 and Rate 2 QOSTBCs, since each transmitted symbol is detected individually. This represents a linear increase for PIC and a polynomial increase for JD, as throughput is increased through constellation size. When n_T increases, PIC maintains a linear growth in detection complexity whereas JD suffers from exponential growth.

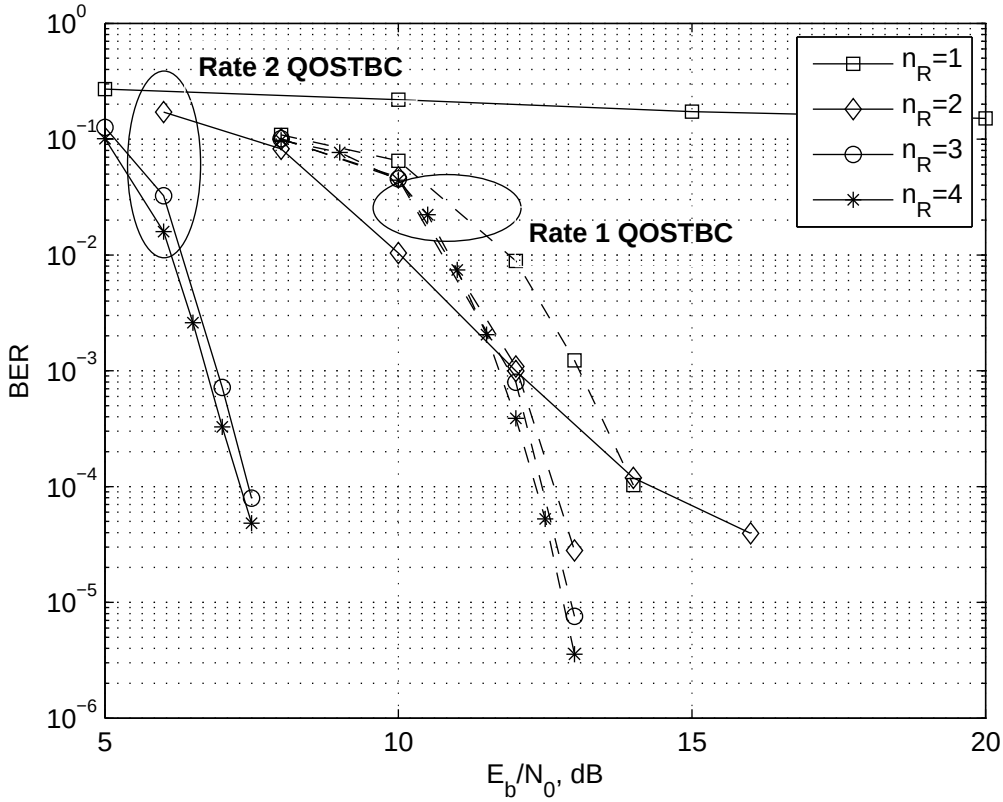


Figure 5.4: BER performance for Rate 1 QOSTBC (dashed lines) and Rate 2 QOSTBC (solid lines) using n_R=1,2,3 and 4. The Rate 1 QOSTBC uses 16QAM and JD, while the Rate 2 QOSTBC uses QPSK and PIC. (343,186) LDPC component codes are used, giving the same system throughput of 2 bits/symbol period.

5.4.3 Number of Required Receive Antennas

OSTBCs were originally designed for a single receive antenna [8, 24]. Although the QOSTBC in [17] was also designed for $n_R = 1$, here we demonstrate that performance can be significantly improved by using $n_R > 1$. Fig. 5.4 shows the bit-error rate (BER) performance for the Rate 1 and Rate 2 QOSTBCs using n_R=1,2,3 and 4. For a fair comparison, the Rate 1 QOSTBC employs JD and 16QAM, while the Rate 2 QOSTBC employs PIC and QPSK. This produces the same approximate throughput of about 2 bits/symbol period. The Rate 1 QOSTBC shows a significant performance gain as we increase n_R from 1 to 2, but minimal gain from 2 to 3 and from 3 to 4. On the other hand, the Rate 2 QOSTBC exhibits significant performance gain as we increase n_R from 1 to 2 and from 2 to 3, but minimal gain from 3 to 4.

This is because the Rate 1 QOSTBC in [17] generates one interference term (see equation (5.4)), and this increases the necessary degrees of freedom by one. As in BLAST, the receiver requires n_R greater than the number of co-channel interferers in order to successfully cancel the interference. Therefore, two receive antennas are required to mitigate the effects of co-channel interference. For the Rate 2 QOSTBC, two interference terms are generated for each symbol estimate (see equation (5.5)). Therefore, three receive antennas are required to minimize the effects of co-channel interference. Although $n_R > 3$ receive antennas provides only minimal gain, $n_R = 4$ will be used throughout the rest of this thesis. This enables us to provide a fair comparison to existing MIMO systems, which often use $n_T = n_R = 4$.

Fig. 5.4 also shows that when there is an insufficient number of receive antennas ($n_R = 1$) to achieve the required number of degrees of freedom, JD outperforms PIC. This is because JD is optimal whereas PIC is sub-optimal. On the other hand, once the required degrees of freedom is achieved ($n_R \geq 3$), iterative PIC easily outperforms JD. This is due to the ability of PIC to iteratively exploit the powerful LDPC codes used in the BMCM structure, compared to the one-off use of the same codes with JD.

5.5 Derivation of Bit Metrics to BP Decoders.

The outputs of the JD or PIC detection blocks are the soft symbol estimates, $\hat{s}_i$, of equations (2.13) and (2.14). The m^{th} BP decoder requires as inputs the soft bit metrics $f^0_{m,i} = P(C^m_i = 0|\hat{s}_i)$ and $f^1_{m,i} = P(C^m_i = 1|\hat{s}_i)$. Optimally, we calculate these from the symbol estimates using

$$f^0_{m,i} = \frac{1}{1 + e^{\lambda^m_i}}, \qquad f^1_{m,i} = \frac{1}{1 + e^{-\lambda^m_i}} \tag{5.6}$$

so that $f^0_{m,i} + f^1_{m,i} = 1$, and we define

$$\lambda^m_i = ln \frac{P(C^m_i = 1|\hat{s}_i)}{P(C^m_i = 0|\hat{s}_i)}. \tag{5.7}$$

The soft estimated symbol from the JD or PIC block can be written in the form $\hat{s}_i = K s_i + n_i$, where K is the coefficient of s_i in equations (2.13) or (2.14), and n_i is the sum of the interference terms and modified receiver noise. We assume that n_i is a complex Gaussian random variable. Denote by $\chi^{m,(0)}_i$ the set of symbols associated

with $C_i^m = 0$ and $\chi_i^{m,(1)}$ as those associated with $C_i^m = 1$. Then, equation (5.7) can be rewritten as

$$\lambda_i^m = ln\frac{\Sigma_{\beta\in\chi_i^{m,(1)}} P(s_i = K\beta|\hat{s}_i)}{\Sigma_{\gamma\in\chi_i^{m,(0)}} P(s_i = K\gamma|\hat{s}_i)} \tag{5.8}$$

By assuming that $P(s_i = K\beta) = P(s_i = K\gamma)$ for $\beta \in \chi_i^{m,(1)}$ and $\gamma \in \chi_i^{m,(0)}$, and applying Bayes rule, we obtain

$$\lambda_i^m = ln\frac{\Sigma_{\beta\in\chi_i^{m,(1)}} P(\hat{s}_i|s_i = K\beta)}{\Sigma_{\gamma\in\chi_i^{m,(0)}} P(\hat{s}_i|s_i = K\gamma)}. \tag{5.9}$$

Since n_i is assumed to be Gaussian, $P(\hat{s}_i|s_i = K\beta) = \frac{1}{\sigma\sqrt{2}}\exp(-\frac{1}{2\sigma^2}||\hat{s}_i - K\beta||^2)$, and hence we can rewrite equation (5.9) as

$$\lambda_i^m = ln\frac{\Sigma_{\beta\in\chi_i^{m,(1)}} \exp(-\frac{1}{2\sigma^2}||\hat{s}_i - K\beta||^2)}{\Sigma_{\gamma\in\chi_i^{m,(0)}} \exp(-\frac{1}{2\sigma^2}||\hat{s}_i - K\gamma||^2)}. \tag{5.10}$$

Using the approximation $ln(\Sigma_j exp(-X_j)) \approx -\min_j(X_j)$, equation (5.10) can be simplified to

$$\lambda_i^m \approx \frac{1}{2\sigma^2}\left[\min_{\beta\in\chi_i^{m,(0)}}||\hat{s}_i - K\gamma||^2 - \min_{\beta\in\chi_i^{m,(1)}}||\hat{s}_i - K\beta||^2\right] \tag{5.11}$$

where σ^2 is the variance of n_i.

It was shown in [71] that omitting the variance from equation (5.11) provides a slight improvement in performance at high SNR. In equation (5.11), λ_i^m is expressed as the difference between two minimum squared Euclidean distances. This distance is simply a distance metric characterized by the L_2-norm in Euclidean space. In [71], this distance metric was generalized to an L_p-norm, given by

$$\lambda_i^m \approx \min_{\gamma\in\chi_i^{m,(0)}}||\hat{s}_i - K\gamma||^p - \min_{\beta\in\chi_i^{m,(1)}}||\hat{s}_i - K\beta||^p \tag{5.12}$$

It was found that decreasing p to 1.9 provided some gain in performance over $p = 2$ [71]. We extend this idea further to $p = 1$ due to the simplicity of calculating the L_1-norm[5]. Then, equation (5.11) takes the form

$$\lambda_i^m \approx \min_{\gamma\in\chi_i^{m,(0)}}||\hat{s}_i - K\gamma||^1 - \min_{\beta\in\chi_i^{m,(1)}}||\hat{s}_i - K\beta||^1 \tag{5.13}$$

where $||.||^1$ denotes the L_1-norm. The calculation of the variance and the squaring operation are not used in equation (5.13), resulting in less computation than equation

[5]In the limit when $p = 0$, λ_i^m effectively reduces to a Hamming distance metric.

(5.11). We have found that this yields good performance at low complexity. Note that including the noise variance at high SNR and using larger values of p has the effect of magnifying unreliable values of λ_i^m and was shown in [71] to degrade performance.

5.6 Stopping Criteria for Iterative PIC-BP Decoding

In Section 5.4.1, we performed one-off JD to get improved symbol estimates, followed by parallel BP decoding on the corresponding bit estimates. The detection and decoding processes were then terminated and the hard outputs from each BP decoder were used to determine the BER. In Section 5.4.2, the hard outputs from each BP decoders were used in one of two ways:
(a) to remodulate the estimated symbol for the next PIC update, or
(b) to determine the BER if no further PIC updates are required.
Deciding when to terminate the PIC update poses an interesting question.

The BP decoding process, between PIC updates, terminates whenever a valid codeword is found or when some number BP_{max} decoding iterations is reached. Assuming that a subsequent PIC update is required, this poses a second question: Do we need to perform BP decoding on the m^{th} codeword, if a valid m^{th} codeword has already been found during the previous BP decoding? The bit metrics sent to the m^{th} BP decoder change after every PIC update because the PIC update is performed using bit estimates from all M BP decoders (through the constellation mapper), not just the m^{th} BP decoder. Therefore, a valid codeword after any PIC update may result in a different valid codeword or even a non-valid codeword if BP decoding is performed after the next PIC update. This may or may not be desirable, depending on whether the original valid codeword is correct.

To attempt to answer the two questions posed above, we consider two different stopping criteria for PIC detection and BP decoding, with varying degrees of decoding complexity. We denote P_{max} as the maximum number of PIC updates to be performed in all cases.

No Stopping Criteria: We perform all P_{max} PIC updates. After each PIC update, we perform BP decoding on all M codewords, disregarding the fact that some valid codewords may have already been found from the previous BP decoding.

Stopping Criterion I: We stop performing PIC updates if all M BP decoders find valid codewords after the previous PIC update, or after P_{max} is reached. If a PIC update was executed, we perform BP decoding on all M codewords, disregarding the fact that some valid codewords may have already been found during the previous BP decoding.

Stopping Criterion II: We stop performing PIC updates if all M BP decoders find valid codewords after the previous PIC update, or after P_{max} is reached. If a PIC update is executed, we only perform BP decoding on the codewords where no valid codewords were found during previous BP decodings. Therefore, BP decoding may not be performed on all M codewords after each PIC update. For codewords where BP decoding is not performed, the outputs of valid codewords from the previous BP decoding are carried forward as outputs of the current BP decoding.

Fig. 5.5 illustrates the differences between all three cases using an example with $M = 4$ parallel BP decoders. A maximum of $P_{max} = 3$ PIC updates are performed. Recall that after linear processing, the soft symbol estimates are passed directly to the demapper and the corresponding bit metrics are calculated for each BP decoder. BP decoding is then performed on all $M = 4$ codewords. For each codeword, BP decoding is performed up to $P_{max} + 1 = 4$ times. In all three cases, immediately after linear processing, we assume as example that a valid codeword is found in BP Decoder 3. This is indicated by a $\star$ symbol next to BP Decoder 3, as shown in Fig. 5.5.

When no stopping criteria are used, we perform the first PIC update. This is followed by BP decoding on all $M = 4$ codewords, resulting in valid codewords being found in BP Decoders 1, 3 and 4. We then perform the second PIC update. This is followed by BP decoding on all $M = 4$ codewords, this time resulting in valid codewords being found in all BP decoders. Then, we perform the third and last

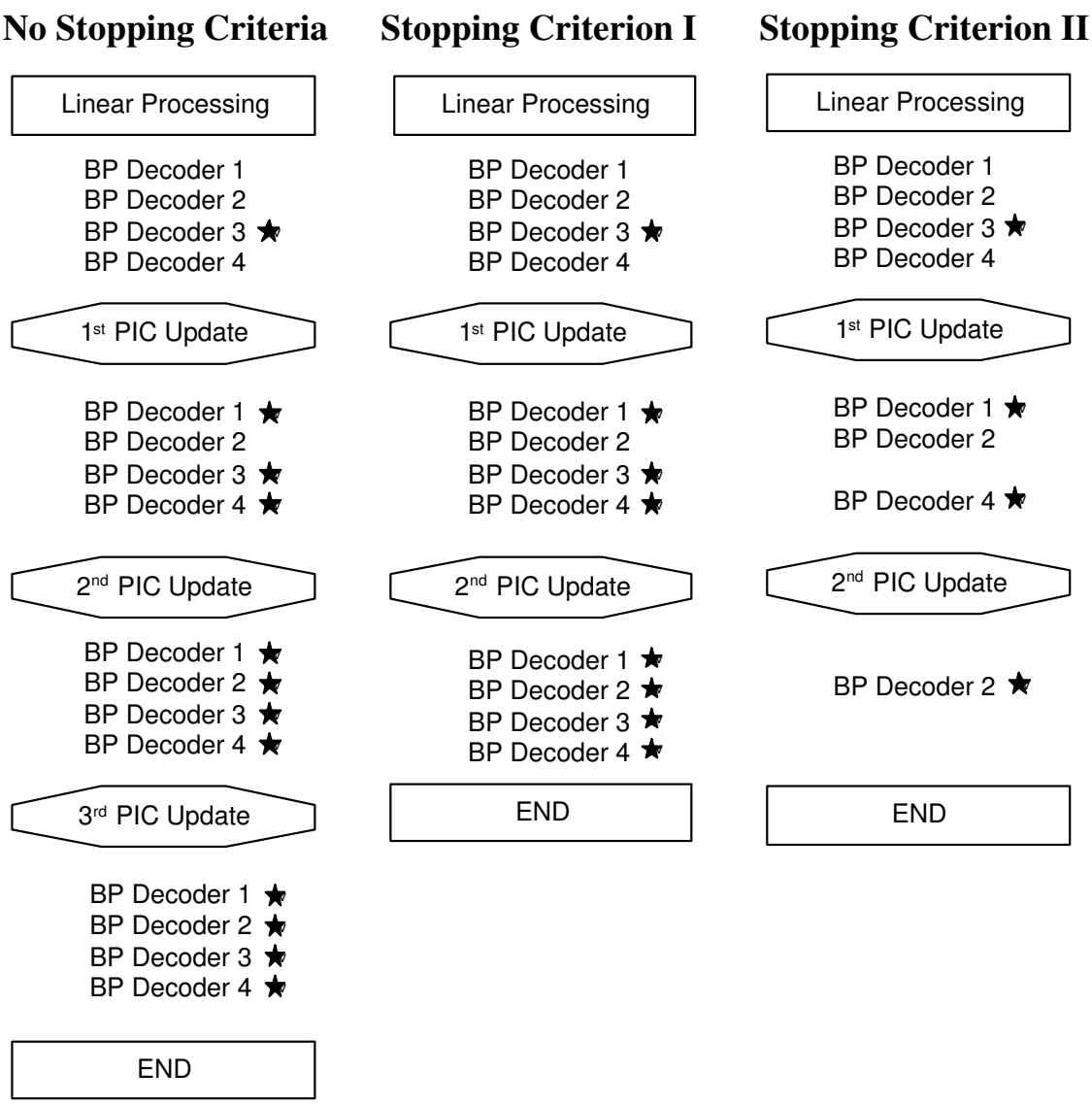


Figure 5.5: How PIC updates and BP decodings are performed when different stopping criteria are used. We use $P_{max} = 3$ and $M = 4$ parallel BP decoders. The $\star$ symbol beside BP decoder m means that a valid codeword was found for the m^{th} BP decoder between PIC updates.

PIC update. This is followed by BP decoding on all codewords, resulting in valid codewords for all BP Decoders. A valid codeword for the m^{th} BP decoder after the previous PIC update does not guarantee that a valid codeword will be found for the m^{th} BP decoder after the current PIC update. Since $P_{max} = 3$ has already been reached, no further PIC update is performed, and the outputs from all BP decoders (which may or may not have converged to a valid codeword) are used to determine the error rate performance.

For Stopping Criteria I, we perform the first PIC update. This is followed by BP decoding on all $M = 4$ codewords, resulting in valid codewords being found in BP Decoders 1, 3 and 4. We then perform the second PIC update. This is followed by BP decoding on all $M = 4$ codewords, this time resulting in valid codewords being found in all BP decoders. Since valid codewords are found for all $M = 4$ BP decoders, we terminate the whole detection-decoding process. The $M = 4$ valid codewords are then used to determine the error rate performance. In this case, P_{max} is not reached.

For Stopping Criteria II, we perform the first PIC update. Since the last BP decoding found a valid codeword in Decoder 3, BP decoding is now only performed in Decoders 1,2 and 4. This results in valid codewords being found in BP Decoders 1 and 4. We then perform the second PIC update. This is followed by BP decoding on only Decoder 2, resulting in a valid codeword. Valid codewords from previous BP decodings are carried forward if no further BP decoding is performed for any particular BP decoder. Therefore, we now have $M = 4$ valid codewords and the whole detection-decoding process terminates. Finally, the error rate performance is determined.

5.7 Summary

In this chapter, we have described the proposed high-throughput BMCM-QOSTBC-PIC MIMO system. The BMCM-QOSTBC scheme allows the system to achieve high throughput. The PIC detection scheme ensures that the system remains practical by offering low detection complexity compared to JD and small delay compared to SIC. Good performance of this proposed high-throughput system can be achieved using the LDPC codes from Chapter 4. Simulation results at various throughputs are provided in Chapter 6 assuming quasi-static flat Rayleigh fading and ideal channel state information (CSI). Comparisons to other MIMO systems are also provided. In Chapter 7, we consider the effects of imperfect CSI at the receiver and investigate the effects of allowing the subchannels to vary continuously with time according to a normalized fade rate, $f_D T$.

Chapter 6

Performance Results for High Throughput Systems

6.1 Introduction

In this chapter, we consider the performance of the high throughput MIMO system using BMCM and iterative PIC-BP decoding which was described in Chapter 5. Simulation parameters and notations are given in Section 6.2. In Section 6.3, we investigate the effects on performance of increasing the number of PIC updates, and the number of BP iterations between PIC updates. For a given processing delay constraint, we find a balance between these two parameters in order to give good performance in Section 6.4. In Section 6.5, we show how the proposed system outperforms other MIMO systems in terms of both coding and diversity gains. Performance comparisons are provided at similar throughputs. In Section 6.6, we show how performance can be further improved by using unequal error protection on 16-QAM or larger constellations. We can further reduce the complexity of the iterative PIC-BP decoding scheme by employing the stopping criteria from Section 5.6. The cost of this complexity reduction on performance is discussed in Section 6.7. Finally, a brief summary of the results in this chapter is provided in Section 6.8.

6.2 Simulation Parameters

All systems considered use $n_T = 4$ transmit and $n_R = 4$ receive antennas[1]. Gray mapped 16-QAM and quaternary phase shift keying (QPSK) constellations are used. We introduce the notation $Modulation - QOSTBC - Detection$ to indicate the different modulation, QOSTBC and detection schemes used for MIMO systems employing BMCM. We use R1 and R2 to denote the Rate 1 QOSTBC and Rate 2 QOSTBC, respectively. For example, the 16QAM-R2-PIC system uses 16-QAM, the Rate 2 QOSTBC and PIC detection.

We use BP_{max} to indicate the maximum number of BP iterations between PIC updates, and P_{max} to indicate the maximum allowed number of PIC updates. The BP algorithm uses an internal stopping criterion which terminates BP decoding, between PIC updates, before BP_{max} iterations if a valid codeword is found. Unless otherwise specified, no stopping criteria from Section 5.6 are used to terminate PIC updates early. The effects on performance of using Stopping Criteria I or II will be presented in Section 6.7. We denote N_{max} as the overall maximum number of BP iterations allowed. Recall from Section 5.4.2 that a PIC update is not performed immediately after linear processing. Therefore, $N_{max} = BP_{max} + P_{max} \times BP_{max}$.

Performance results using Monte Carlo simulations are plotted as frame error rate (FER) or bit error rate (BER) against post-processing SNR. Post-processing SNR is defined at the output of the linear processing block in Fig. 5.3. A discussion of the use of this SNR definition instead of the SNR per receive antenna is given in Appendix A, along with a derivation of the noise variance used. For uncoded systems, a frame is the length of the STBC block, L, while for coded systems, a frame is of length $\lceil \frac{N}{R_{stbc}} \rceil$, where N is the block length of the LDPC component codes and $\lceil x \rceil$ denotes the smallest integer greater than[2] x. Each simulation point contains at least 100 frame errors. A frame is considered to be in error if and only if any of its data bits are in error.

The total transmit power across all $n_T = 4$ antennas is normalized to 1 for each time slot. We assume the $n_T n_R$ subchannels to be independent. In this chapter,

[1]The choice of $n_R = 4$ is discussed in Section 5.4.3.

[2]The LDPC component codes are zero-padded to ensure that the coded frame length is an integer.

quasi-static fading is assumed on each subchannel, where the subchannel remains constant for each STBC block of L time slots, and varies independently from block to block (used for both uncoded and coded cases). Perfect CSI is assumed to be available at the receiver, but none is available at the transmitter.

6.3 Increasing P_{max} and BP_{max}

Fig. 6.1 illustrates coded BER performance for the proposed 16QAM-R2-PIC system, using four (343,186) LDPC component codes in a BMCM structure. It shows performance after 1, 2, 4, 5, 9 and 19 PIC updates. The BP decoders use a maximum of $BP_{max} = 20$ iterations between PIC updates. The notation $BP = 20(4)$ indicates that a maximum of 20 BP iterations are used between PIC updates and 4 PIC updates are used in total. This means a maximum of $N_{max} = 100 = 20 + 4 \times 20$ BP iterations are used. The performance gain after more than 5 PIC updates is minimal. The BER curve after 4 PIC updates is falling at almost 2 decades/dB[3] and is not showing any signs of an error floor at 10^{-6}. Each additional PIC update, coupled with BP decoding, provides diminishing improvement in performance.

Fig. 6.2 illustrates performance for the same 16QAM-R2-PIC system and shows the effect of increasing BP_{max} from 1 to 40 while fixing $P_{max} = 9$. The uncoded BER performance is also shown. When no FEC coding is used, iterative PIC detection (also with $P_{max} = 9$) shows a very high error floor, around BER$=2 \times 10^{-2}$. The iterative PIC algorithm on its own converges with a substantial amount of co-channel interference in the symbol estimates. This interference effect is reduced by using FEC coding.

When FEC coding is used in the form of LDPC codes together with joint iterative PIC-BP decoding, substantial coding and diversity gains are obtained, for $BP_{max} = 1$ and $P_{max} = 9$. Increasing BP_{max} from 1 to 5 provides diminishing gains in performance. Using more BP iterations generally provides better decoded bit estimates. However, a slight coding loss is observed when BP_{max} is increased from 5 to 10. This is because the BP decoders are allowed sufficient iterations between PIC updates to converge to valid codewords. However, many of these valid codewords

[3]We measure the fall rate of the performance curves using the intuitive $\frac{\text{vertical difference}}{\text{horizontal difference}}$ ratio, resulting in units of decades/dB.

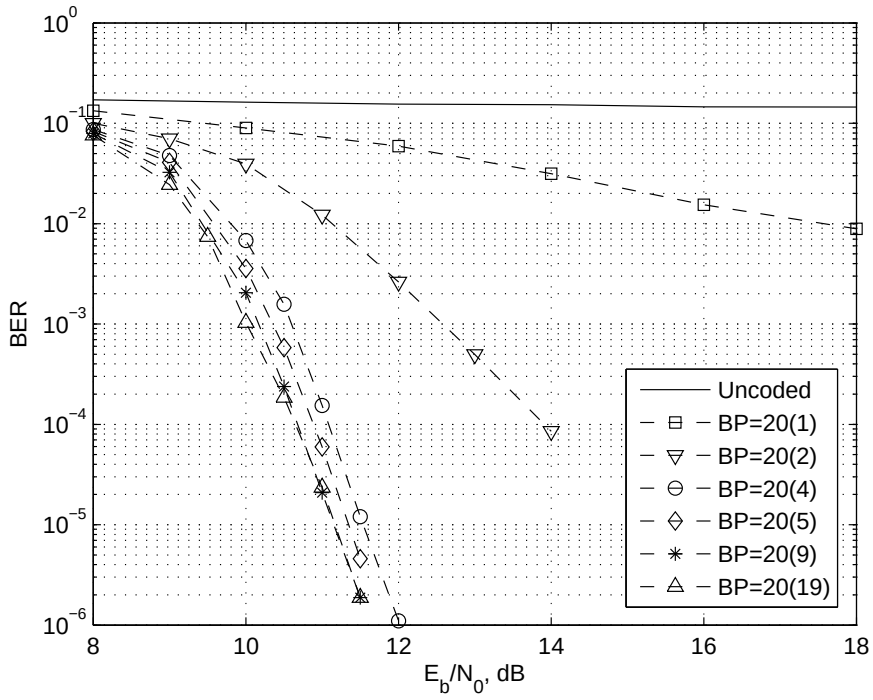


Figure 6.1: BER performance of the proposed 16QAM-R2-PIC system using four (343,186) LDPC component codes for $P_{max} = 1, 2, 4, 5, 9, 19$ and $BP_{max} = 20$. Throughput is 4.3 bits/symbol period.

are actually incorrect codewords. This results in a larger number of incorrect bits for $BP_{max} = 10$, compared to the corresponding number for $BP_{max} = 5$ as in this case BP decoding is terminated before an incorrect codeword appears.

When BP_{max} is increased further from 10 to 40, even more valid codewords are found, but an increasing percentage of these are correct codewords. As a result, diminishing improvements in performance are once again observed.

Note that at high SNR, no improvement in diversity gain is provided after about $BP_{max} = 2$. The performance when $BP = 3(9)$ in Fig. 6.2 is significantly better than the performance when $BP = 20(1)$ in Fig. 6.1, even though N_{max} is higher in the latter. These results demonstrate the importance of performing a sufficient number of PIC updates in order to achieve maximum diversity gain.

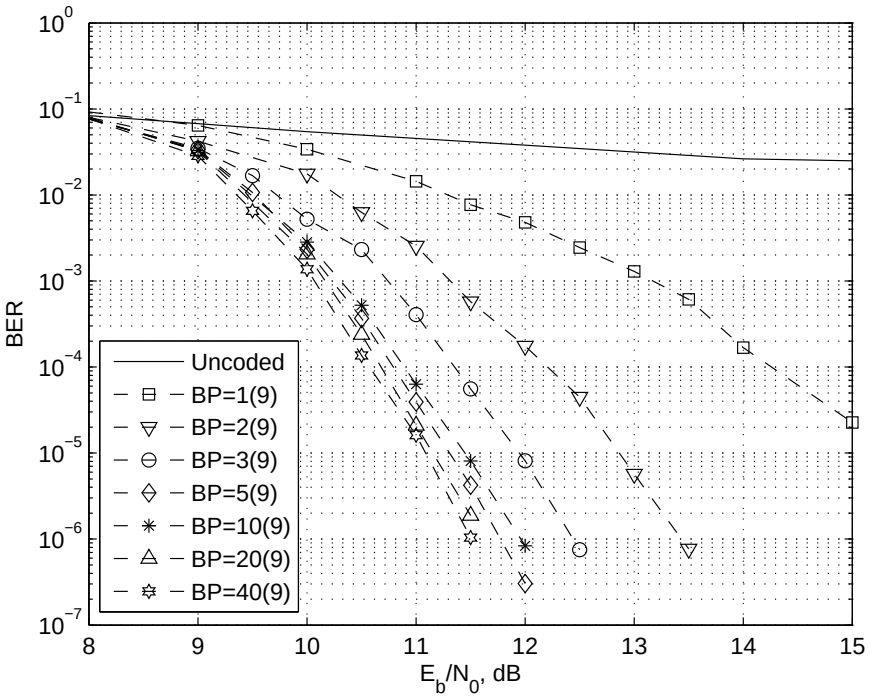
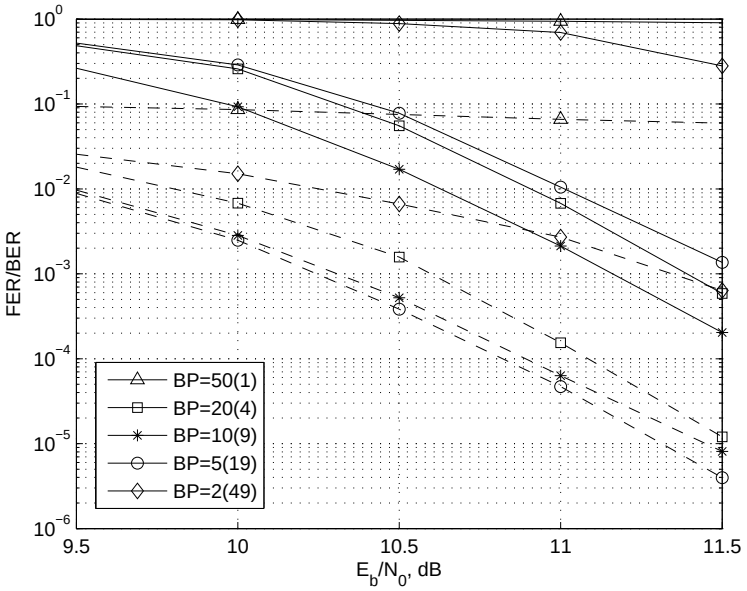


Figure 6.2: BER performance of the proposed 16QAM-R2-PIC system using four (343,186) LDPC component codes for $BP_{max} = 1, 2, 3, 5, 10, 20, 40$ and $P_{max} = 9$. Throughput is 4.3 bits/symbol period.

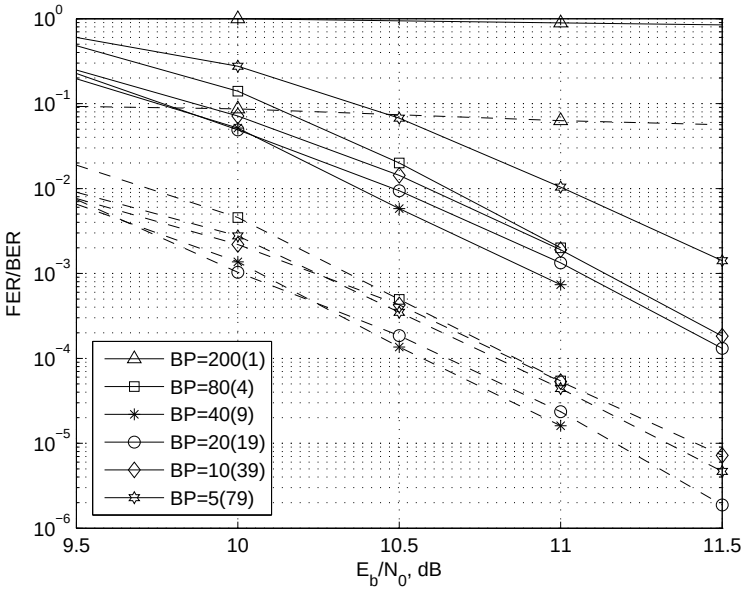
6.4 How often should we perform PIC updates?

Although both PIC detection and BP decoding are considered low complexity schemes, both these processes inevitably introduce small delays. In certain systems such as those requiring voice transmission, a minimal time delay is critical to the quality of the transmission. One of the contributors to the delay in our systems is the overall maximum number of BP iterations used, N_{max}. In this section, we investigate the frequency of PIC updates required to give the best performance, when N_{max} is fixed.

Fig. 6.3 shows the frame error rate (FER) and BER performance of the 16QAM-R2-PIC scheme using (343,186) LDPC component codes, providing a throughput of 4.3 bits/symbol period. Performance is shown for different numbers of PIC updates and BP iterations, when N_{max} is fixed at 100 and also at 400. Both FER and BER performances generally improve as the number of PIC updates is increased. For $N_{max} = 100$, increasing the number of PIC updates beyond 9 and 19 degrades FER and BER performance, respectively, because the number of allowable BP it-



(a) $N_{max} = 100$



(b) $N_{max} = 400$

Figure 6.3: FER (solid lines) and BER (dashed lines) performance of the 16QAM-R2-PIC BMCM system using (343,186) LDPC component codes. Performance is shown for different numbers of PIC updates when N_{max} is fixed. Throughput is 4.3 bits/symbol period.

erations between PIC updates decreases to a level such that the BP algorithm is unable to converge properly. Similar trends are observed after 9 PIC updates, for $N_{max} = 400$.

To obtain good performance for any fixed N_{max}, we have found that heuristically setting $BP = \sqrt{N_{max}}(\sqrt{N_{max}} - 1)$ provides a good starting point. This corresponds to a ratio of P_{max} to BP_{max} of approximately 1. From here, we can increase or decrease this ratio to find the best performance for a given N_{max}. For $N_{max} = 100$, the best FER and the second best BER performances are produced by setting $BP = 10(9)$, as shown in Fig. 6.3a. For $N_{max} = 400$, the best FER and BER performances are produced by setting $BP = 40(9)$, whereas the second best performance in each case is given by $BP = 20(19)$, as shown in Fig. 6.3b.

6.5 Comparison to other MIMO systems

Fig. 6.4 shows the comparative performances of different MIMO systems. The modulation schemes are chosen to result in the same approximate throughput of 2 bits/symbol period in all cases. The uncoded Rate 1/2 OSTBC of [24] using linear processing (LP) has the worst performance in the group because it pays a big penalty due to the low value of R_{stbc}, requiring the use of 16-QAM and no LDPC codes to obtain the required throughput. It attains a BER of 10^{-4} at 19.2dB. The Rate 1 QOSTBC of [17] using pairwise joint detection (JD) reaches the same BER at 15.7dB, a gain of 3.5dB over the OSTBC. This is primarily because the Rate 1 QOSTBC has double the rate and so we can use a smaller constellation (QPSK) with a larger minimum Euclidean distance between constellation points.

We also compare the performance in Fig. 6.4 of the coded 16QAM-R1-JD and 16QAM-R1-PIC schemes using four (343,186) LDPC component codes. When JD is used, the symbol estimates from the JD block are demapped to bits and the bit estimates are passed to the parallel BP decoders. A maximum of 200 BP iterations is used to decode each LDPC code. Due to the high complexity of the JD detection scheme, JD is performed only once. When PIC is used, the symbol estimates on the first iteration are passed directly to the BP decoders after linear processing. On subsequent iterations, information from the BP decoders is passed to

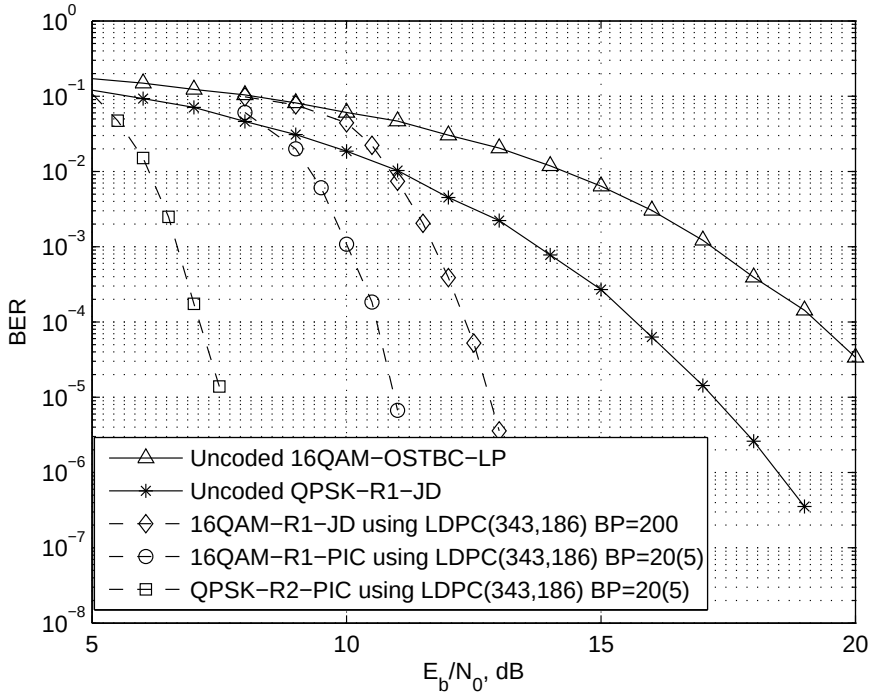


Figure 6.4: BER performance comparison of coded and uncoded systems for the same approximate throughput of 2 bits/symbol period.

the PIC block. The coded 16QAM-R1-JD scheme achieves a BER of 10^{-4} at 12.4dB, a gain of 3.3dB over the uncoded QPSK-R1-JD scheme. The coded 16QAM-R1-PIC scheme achieves the same BER at 10.7dB, a further gain of 1.7dB over the coded 16QAM-R1-JD scheme. Although JD is optimal in terms of symbol detection, it it not optimal over the whole codeword. Hence, iterative PIC is able to outperform one-off JD.

When the Rate 2 QOSTBC is used in conjunction with the same LDPC component codes and PIC, QPSK yields the required throughput. Fig. 6.4 shows that this scheme achieves a BER of 10^{-4} at about 7.2dB, a gain of 3.5dB over coded 16QAM-R1-PIC. Therefore, we find that it is better to increase throughput by increasing R_{stbc} than by increasing constellation size. The cost is an increase in detection complexity due to co-channel interference. The QPSK-R2-PIC system also has an overall gain of 12dB over the uncoded Rate 1/2 OSTBC.

We also consider the Rate 3/4 super-OSTBC (SOSTBC) of [18]. The definition of E_b/N_0 and total transmit energy are different in [18]. Therefore, we scale their

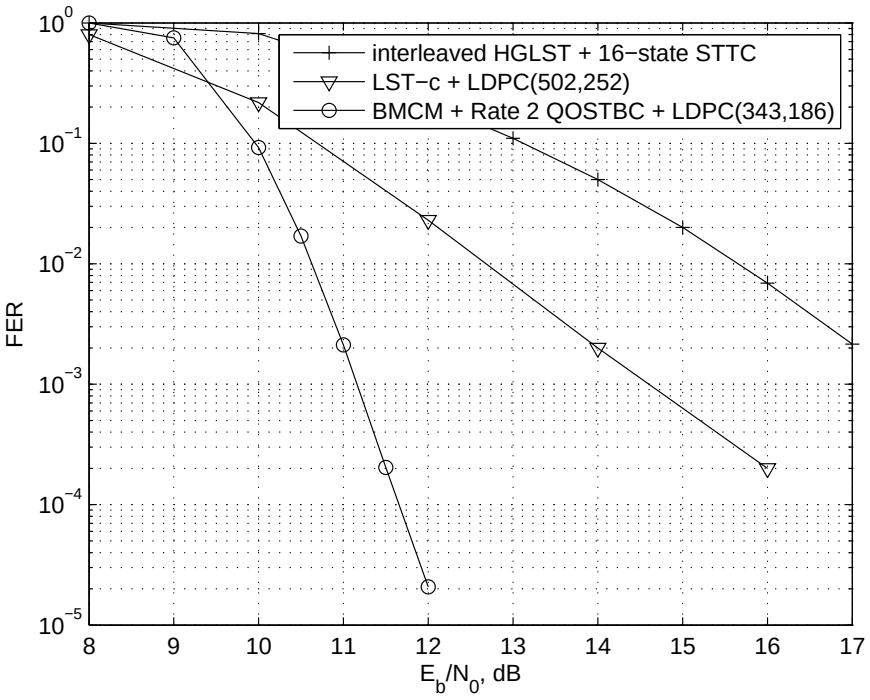


Figure 6.5: FER performance comparison between our proposed system (4.3 bits/symbol period), the interleaved HGLST system (4 bits/symbol period) of [66], and the LST-c system (4 bits/symbol period) of [67].

results to normalize the total transmit energy to our definition and redefine E_b/N_0 to be the post-processing SNR. We then find that the uncoded SOSTBC follows the performance curve of the Rate 1/2 OSTBC, with a 0.5dB SNR improvement. This is achieved using an 8-PSK constellation, giving it a slightly higher throughput of 2.25 bits/symbol period.

Fig. 6.5 compares the FER performance of our 16QAM-R2-PIC system (4.3 bits/symbol period) to that of the interleaved horizontal generalized layered space-time (HGLST) system[4] of [66] (4 bits/symbol period) and the LST-c layered space-time system[5] of [67] (4 bits/symbol period). The FER curves from [66] and [67] have been shifted to the right by 6dB due to the difference in SNR definitions.

The interleaved HGLST scheme uses a 16-state space-time trellis code, 130 time slots in a frame and ordered SIC at the receiver. The LST-c scheme transmits

[4]This system was chosen due to its comparable throughput, hybrid space-time code structure and comparable values of n_T and n_R.

[5]This system was chosen due to its comparable throughput, similar detection-decoding scheme and comparable values of n_T and n_R.

QPSK symbols directly using interleaving over both space and time, instead of using space-time codes. It uses a (504,252) LDPC component code on each transmit antenna, giving a frame length of 252 time slots. The LST-c scheme uses a modified PIC detector [69], that uses decision statistics combining to improve performance at the cost of increased complexity. Interleaving and deinterleaving are performed between the PIC detector and n_T BP decoders in an iterative fashion. On the other hand, our BMCM-QOSTBC-PIC system employs a standard PIC detector, which is computationally less complex than either the modified PIC detector of [69] or ordered SIC. Constellation mapping and demapping are performed between the PIC detector and the M BP decoders. We use (343,186) LDPC component codes and 172 time slots in a frame.

As shown in Fig. 6.5, our BMCM-QOSTBC-PIC system outperforms the HGLST and LST-c systems by about 4.5dB and $1.6dB$, respectively, at a FER of 10^{-2}. Our FER curve is falling at 2 decades/dB, compared to only 0.5 decade/dB for the other two systems. Therefore, we achieve greater diversity gain, despite using a lower complexity detection scheme.

6.6 Unequal Error Protection Using 16-QAM

In Gray-mapped 16-QAM systems, two of the four bits that label the constellation points are better protected than the other two, due to a larger average distance of each constellation point to the decision boundary corresponding to each bit [72]. This is illustrated in Fig. 6.6 where a Gray-mapped 16-QAM constellation (denoted constellation ABCD) is divided according to its four sub-labels - A,B,C and D (denoted constellations A,B,C and D, respectively). Let d be the minimum distance between any two constellation points. There is one decision boundary in constellations A and B, and two decision boundaries in constellations C and D. It is trivial to show that the average distance of each constellation point to the closest decision boundary, $d_{bd,av}$, is d for constellations A and B, and $0.5d$ for constellations C and D.

Fig. 6.7 shows the BER performance for the 16QAM-R2-PIC system using length 343 LDPC component codes of different rates. In Fig. 6.7a, the (343,186)

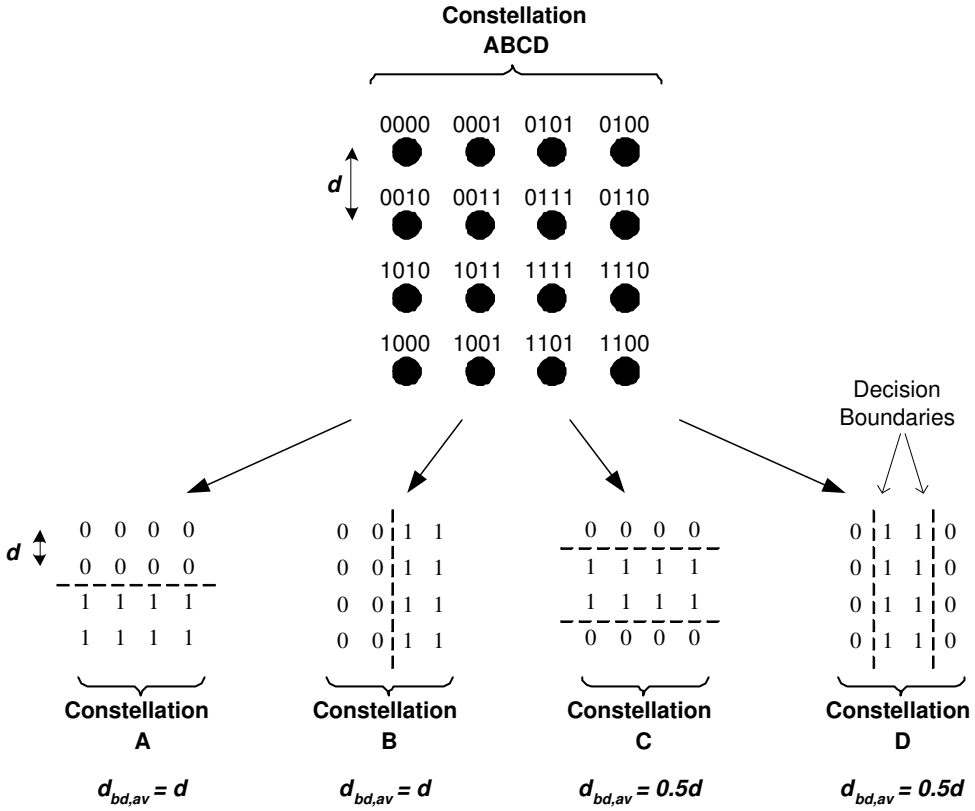
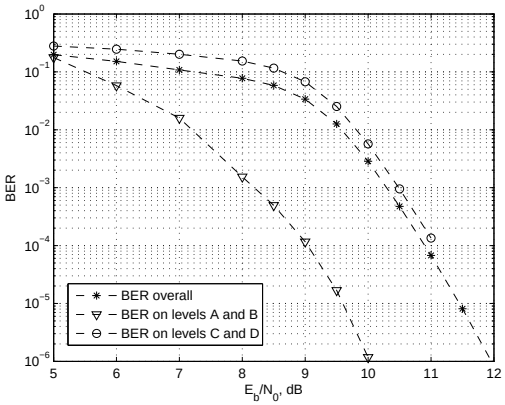


Figure 6.6: Decision boundaries for each sub-label constellation of Gray-mapped 16-QAM.

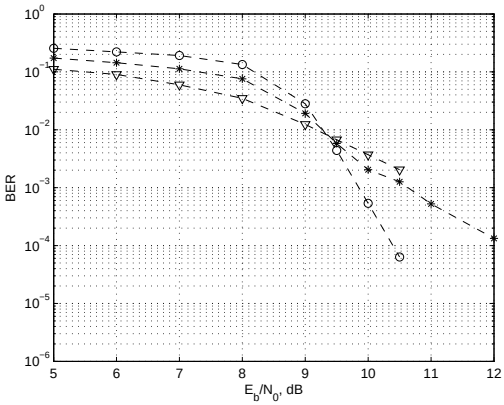
LDPC code is used on all four sub-labels, giving an overall system throughput of 4.3 bits/symbol period. The BER for AB[6] is approximately 2dB better than that on CD. Since $d_{bd,av}$ is larger in AB, these sub-labels have superior error protection. The BERs on AB and CD both fall at around 2 decade/dB. The overall BER reaches 10^{-4} at 10.9dB. However, it is dominated by the performance of CD. This indicates that the superior error performance in AB is wasted. Hence, we can increase the code rate on these sub-labels, thereby increasing the overall system throughput.

Fig. 6.7b uses the (343,252) LDPC code on AB and the (343,186) LDPC code on CD, increasing the system throughput to 5.1 bits/symbol period. However, the overall BER now reaches 10^{-4} at 12.2dB. This is because the higher rate (343,252) LDPC has inferior error-correction capability, compared to the (343,186) LDPC code. At a BER of 10^{-2}, this system provides a coding gain and an increased

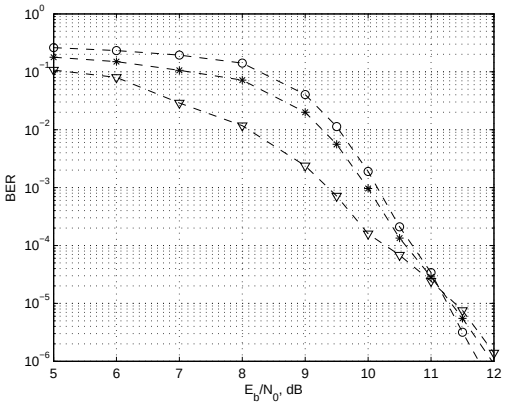
[6]From herein, we use the compact notation AB to denote sub-labels A and B, and CD to denote sub-labels C and D.



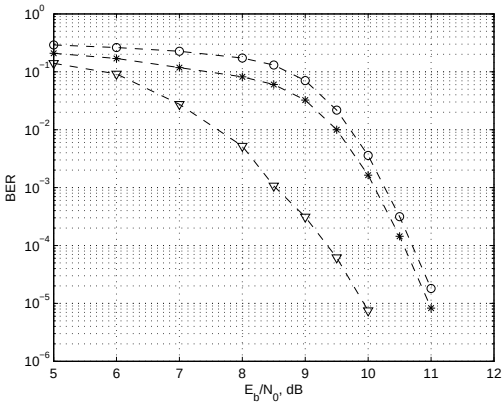
(a) (343,186) LDPC code on AB and (343,186) LDPC code on CD. Throughput is 4.3 bits/symbol period.



(b) (343,252) LDPC code on AB and (343,186) LDPC code on CD. Throughput is 5.1 bits/symbol period.



(c) (343,216) LDPC code on AB and (343,186) LDPC code on CD. Throughput is 4.7 bits/symbol period.



(d) (343,186) LDPC code on AB and (343,156) LDPC code on CD. Throughput is 4.0 bits/symbol period.

Figure 6.7: Performance of the 16QAM-R2-PIC system showing overall BER, BER on sub-labels AB and BER on sub-labels CD.

throughput over that in Fig. 6.7a. For lower overall BER, the performance is now dominated by AB. Although we still have increased throughput, we now have a coding loss of 1.3dB at BER=10^{-4} compared to the system in Fig. 6.7a. The trick here is to ensure that the BER on AB does not cross over the BER on CD until we have reached the desired overall BER.

Fig. 6.7c uses the (343,216) LDPC code on AB and the (343,186) LDPC code on CD, giving a system throughput of 4.7 bits/symbol period. The (343,216) LDPC code has both a code rate and an error-correction capability that lie between the (343,186) and the (343,252) LDPC codes. The crossover between the BER on AB and BER on CD now occurs at around 10^{-5}. The overall BER reaches 10^{-4} at 10.6dB. This represents a coding gain of 0.3dB and an increased throughput of 16% over the system in Fig. 6.7a.

Instead of varying the rate on AB, we can also vary the rate on CD. In Fig. 6.7d, we use the (343,186) LDPC code on AB and a more powerful but lower rate (343,156) LDPC code on CD, giving a system throughput of 4.0 bits/s/Hz. The overall BER now reaches 10^{-4} at 10.6dB. This represents a coding gain of 0.3dB but a loss of throughput by 8% over the system in Fig. 6.7a.

To summarize, we can use more powerful codes on CD to improve the overall BER performance, for a small loss in system throughput. This is because more powerful codes usually have lower code rates, for the same blocklength. A smarter approach would be to increase the system throughput by using higher rate codes on AB. If we choose different component codes with the right balance between code rate and error-correction capability, it is possible to achieve both a coding gain and an increased throughput over systems employing the same component code on all sub-labels, when the sub-labels have different error protection.

Fig. 6.8 shows the BER performance for the 16QAM-R2-PIC system using the (343,186) LDPC code on CD and length 343 LDPC codes with different rates on AB. The code rates on AB vary from 0.54 to 0.64, giving a throughput range of 4.3 to 5.1 bits/symbol period. In Chapter 4, it was shown that higher rate LDPC codes generally have poorer error correction capability, for the same block length. This helps to explain the appearance of error floors at high SNR as we increase the code rate on AB.

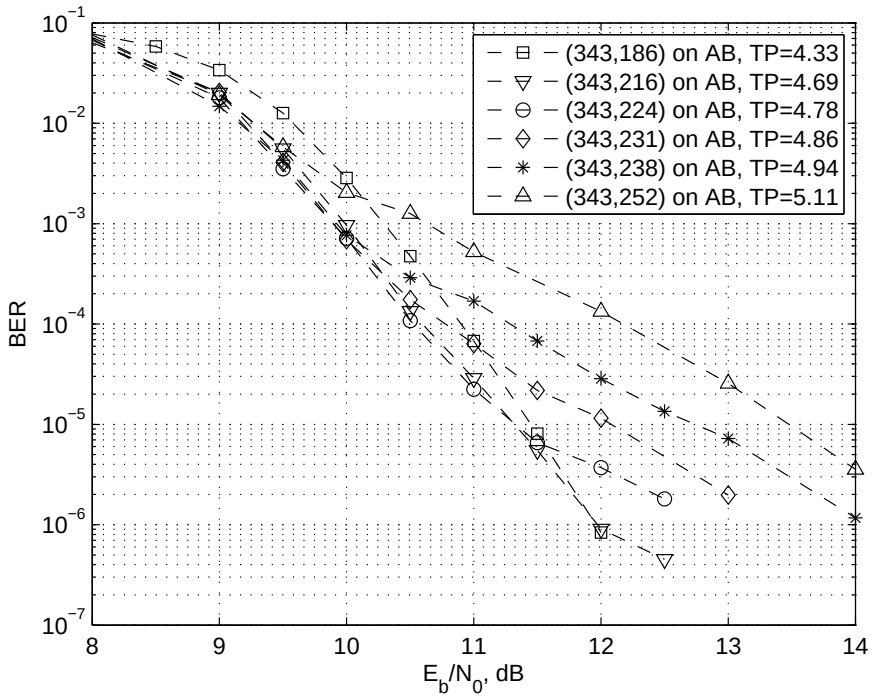


Figure 6.8: BER performance for 16QAM-R2-PIC system using the (343,186) LDPC code on CD, and length 343 LDPC code with different rates on AB. Throughput range is 4.3-5.1 bits/symbol period.

Table 6.1 summarizes the highest throughput achieved for the 16QAM-R2-PIC system in Fig. 6.8 when different BERs are required, within an SNR margin of less than 0.2dB. As the required BER is lowered from 10^{-2} to 10^{-6}, the maximum throughput decreases from 5.1 to 4.7 bits/symbol period. This is because using higher rate LDPC codes on AB causes higher error floors on AB, which dominate the overall BER performance at high SNR. Therefore, the flexibility of the BMCM scheme allows us to tailor different LDPC component codes on sub-labels with different error protection. This allows us to increase throughput when the required BER is raised, without sacrificing performance.

So far, we have only used LDPC component codes of length 343. In Chapter 4, it was shown that longer LDPC codes achieve better BER performance and they also have higher code rates (hence higher throughputs). Fig. 6.9 shows the BER performance for the 16QAM-R2-PIC system using the same component codes of length 2197 on all sub-labels, giving a throughput range of 5.6-6.3 bits/symbol period. As the code rate increases, the error correction capability of the LDPC code decreases

Required BER	Highest Throughput	SNR range	LDPC code on AB
10^{-2}	5.1	9.1-9.3	(343,252)
10^{-3}	4.9	9.9-10.0	(343,238)
10^{-4}	4.8	10.5-10.6	(343,224)
10^{-5}	4.8	11.3-11.4	(343,224)
10^{-6}	4.7	11.9-12.0	(343,216)

Table 6.1
Highest throughput for the 16QAM-R2-PIC system for different required BERs.

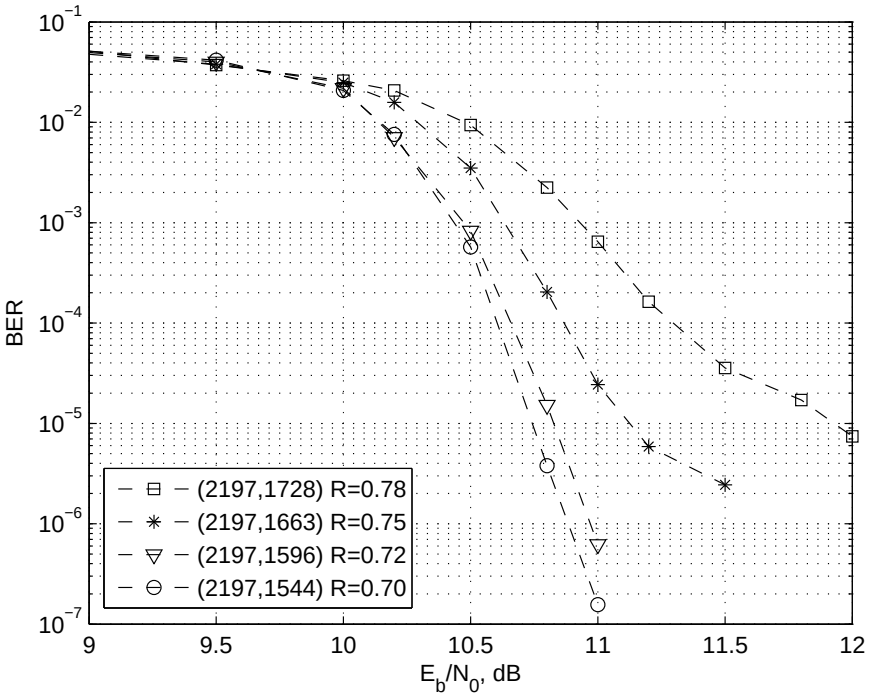


Figure 6.9: BER performance for 16QAM-R2-PIC system using the same component codes of length 2197 on all sub-labels. Throughput (TP) range is 5.6-6.3 bits/symbol period.

and its BER performance degrades. The (2197,1596) and (2197,1544) LDPC codes show no signs of error floors down to BER=10^{-6} and 10^{-7}, respectively. Morever, the (2197,1544) LDPC codes achieve excellent performance at a throughput of 5.6 bits/symbol period with both the FER and BER falling at around 7 decades/dB.

We now focus on the 16QAM-R2-PIC system using the (2197,1663) LDPC component codes on all sub-labels, giving a system throughput of 6.1 bits/s/Hz.

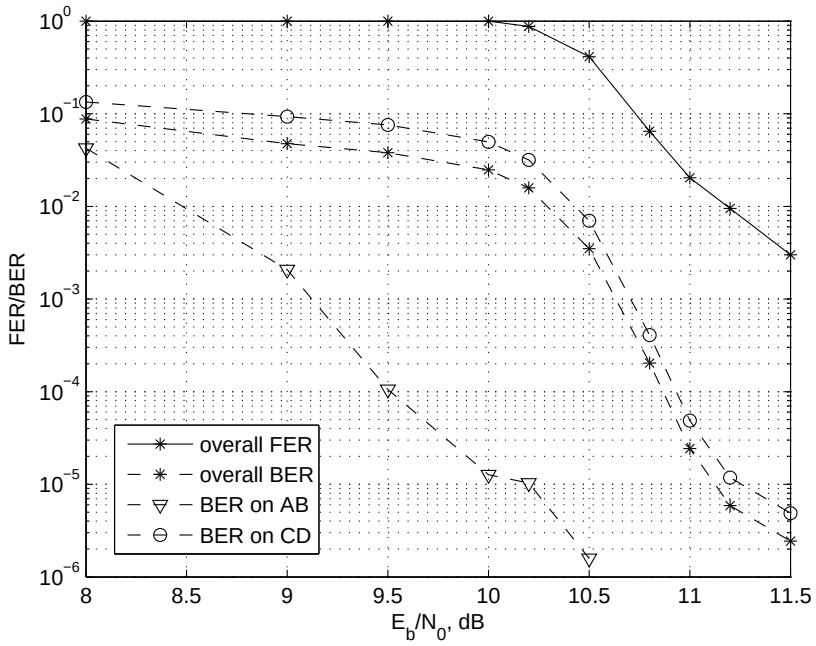
For simplicity, we call this System I. Fig. 6.10a shows the overall FER and BER performance of this system. The BER corresponding to sub-labels AB and CD are shown separately from the overall BER. The overall BER is dominated by sub-labels CD due to their smaller values of $d_{bd,av}$. Therefore, since the (2197,1663) LDPC code on CD has an error floor at about BER=10^{-5}, the overall BER for System I also has an error floor at around the same BER.

Fig. 6.10b shows the same system but using the (2197,1728) LDPC code on AB and (2197,1596) LDPC code on CD, giving the same system throughput of 6.1 bits/symbol period. We call this System II. Although the (2197,1728) LDPC code on AB exhibits an error floor above BER=10^{-5}, the overall BER is still dominated by sub-labels CD which does not exhibit an error floor at this point. Therefore, the overall error floor for System II appears at a lower BER, compared to the error floor for AB. Basically the overall performance is dominated by CD up to E_b/N_0=10.7dB, at which point it becomes dominated by AB.

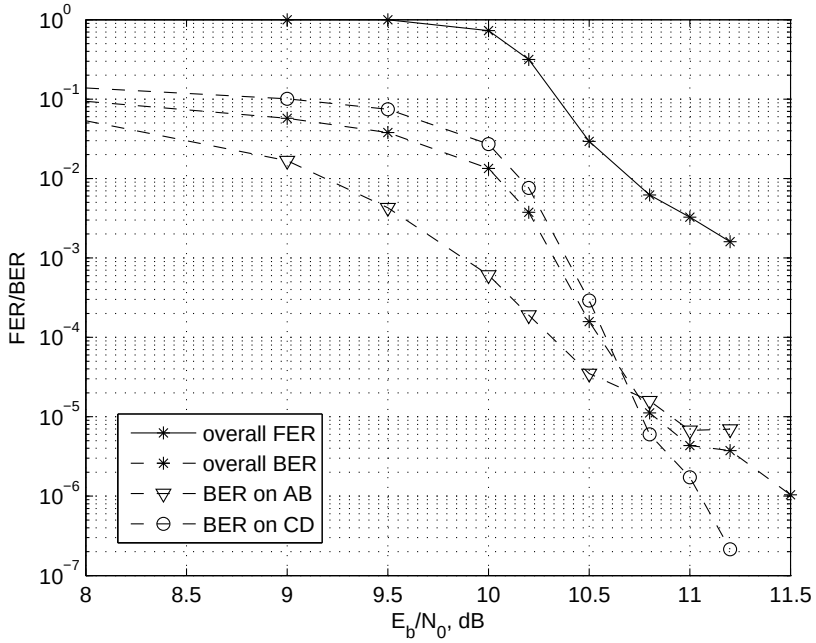
The advantages of using codes with unequal error correction capabilities on sub-labels with unequal error protection are again illustrated in Fig. 6.11. Systems I and II have approximately the same $R_{ldpc} = 0.756$. However, by using a more powerful LDPC on CD and a higher rate LDPC on AB, System II achieves a 0.3dB gain over System I. In addition, System II exhibits error floors at a slightly lower BER than System I. Both these systems have throughput of 6.1 bits/symbol period.

In Fig. 6.11, System III uses a more powerful but lower rate (2197,1702) LDPC code on sub-labels AB, compared to the (2197,1728) LDPC code used on sub-labels AB in System II. This decreases the system throughput by 0.8%, but the error floor has been lowered compared to System II. With a throughput of 6.0 bits/symbol period, System III also has a 0.6dB coding gain over the 16QAM-R2-PIC system using the best[7] combination of length 343 LDPC codes (c.f. Fig. 6.8) at BER=10^{-5}. This also represents a throughput increase of 25%. However, we have increased the frame length from 172 to 1100.

[7]The best combination at BER=10^{-5} uses (343,224) LDPC codes on AB, and (343,186) LDPC codes on CD, giving a throughput of 4.8 bits/symbol period.



(a) Using (2197,1663) LDPC codes on all sub-labels, ABCD.



(b) Using (2197,1728) LDPC codes on AB and (2197,1596) LDPC codes on CD.

Figure 6.10: FER and BER performance of 16QAM-R2-PIC systems. Throughputs are 6.1 bits/symbol period.

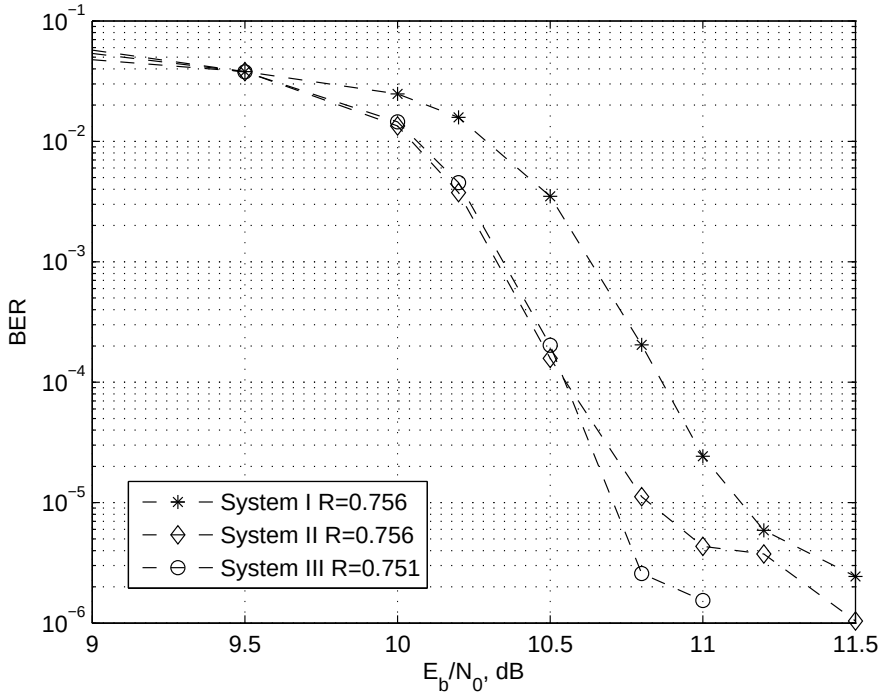


Figure 6.11: BER performance comparison between 16QAM-R2-PIC systems. System I uses the same (2197,1663) LDPC code on all sub-labels, giving $R_{ldpc} = 0.756$. System II uses (2197,1728) LDPC codes on AB and (2197,1596) LDPC codes on CD, giving $R_{ldpc} = 0.756$. System III uses (2197,1702) LDPC codes on AB and (2197,1596) LDPC codes on CD, giving $R_{ldpc} = 0.751$.

6.7 Using Stopping Criteria

In Section 5.6, we introduced two stopping criteria to further reduce the complexity of iterative joint PIC-BP decoding. Under both criteria, we terminate this iterative decoding process when a valid codeword is found by all the M BP decoders. Under Stopping Criterion I, BP decoding is performed on all M sub-labels after every PIC update. Termination only occurs when all M codewords found after the same PIC update are valid. Under Stopping Criterion II, BP decoding is only performed on the sub-labels which did not find a valid codeword during previous PIC updates. Termination occurs when all M valid codewords have been found, which may not happen after the same number of PIC updates.

In this section, we show the effects of employing Stopping Criterion I and then Stopping Criterion II. Finally we compare performance against previous results

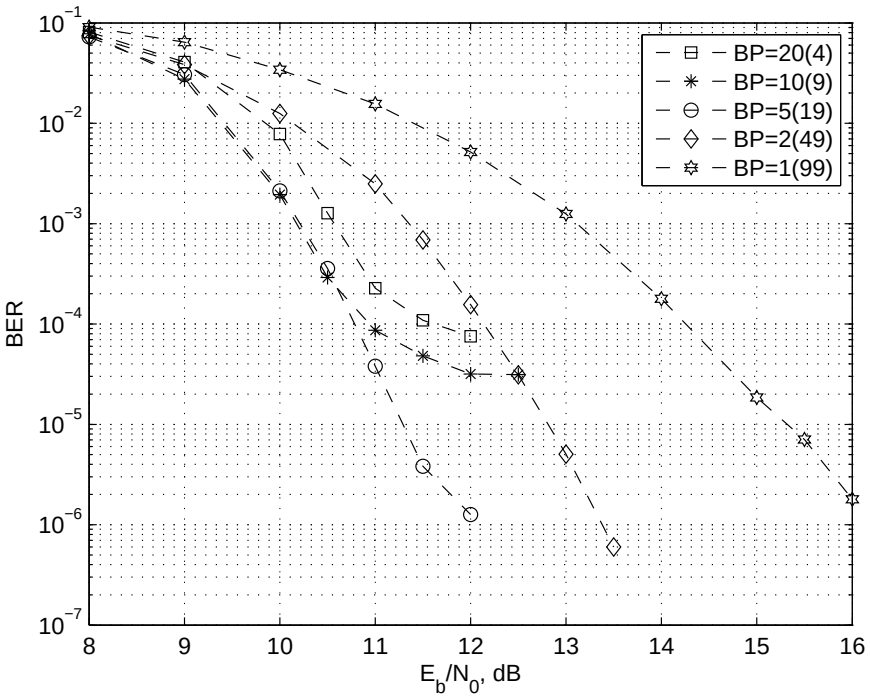


Figure 6.12: BER performance for 16QAM-R2-PIC system using Stopping Criteria I, for $N_{max} = 100$. Throughput is 4.3 bits/symbol period.

where no stopping criteria was used. We divide our observations roughly into two regions - low SNR (below 10.5dB) and high SNR (above 10.5dB).

6.7.1 Stopping Criterion I

Fig. 6.12 shows the effect of using Stopping Criteria I on the BER performance of the 16QAM-R2-PIC system using different values of BP_{max} and P_{max}, for fixed $N_{max} = 100$. At low SNR, performance improves as P_{max} is increased from 4 to 9. However, as P_{max} is increased from 19 to 99, performance starts to degrade because BP_{max} becomes too small for the BP decoder to converge properly.

At high SNR, setting BP=$BP_{max}(P_{max})$=20(4) results in an error floor at BER=7×10^{-5}. Setting BP=10(9) results in an error floor at BER=3×10^{-5}. An error floor is starting to appear at BER=4×10^{-6} when BP=5(19) is used. No error floors are visible down to BER=10^{-6} when BP=2(49) and BP=1(99) are used. These observations show that when the ratio of BP_{max} to P_{max} is high, so is the error floor.

6.7.2 Stopping Criterion II

Fig. 6.13 shows the effect of using Stopping Criterion II on the BER performance of the 16QAM-R2-PIC system using different values of BP_{max} and P_{max}, when $N_{max} = 100$. As in the case with Stopping Criteria I, performance at low SNR improves as P_{max} is increased from 4 to 9 and degrades as P_{max} is increased from 19 to 99. At high SNR, it is again clear that the ratio of BP_{max} to P_{max} determines the error floor level. The importance of this ratio raises two questions: (a) How does P_{max} affect performance? (b) How does BP_{max} affect performance?

Fig. 6.14 shows the effect on both the FER and BER performance when P_{max} is increased from 5 to 19, for fixed BP_{max}=10. At low SNR, increasing the number of PIC updates provides diminishing coding gains. When the error floor is reached at high SNR, additional PIC updates have no effect on performance.

Fig. 6.15 shows the effect on both the FER and BER performance when BP_{max} is increased from 2 to 20, for fixed P_{max}=9. At low SNR, increasing BP_{max} from 2 to 10 provides diminishing gain in the FER. Increasing BP_{max} from 2 to

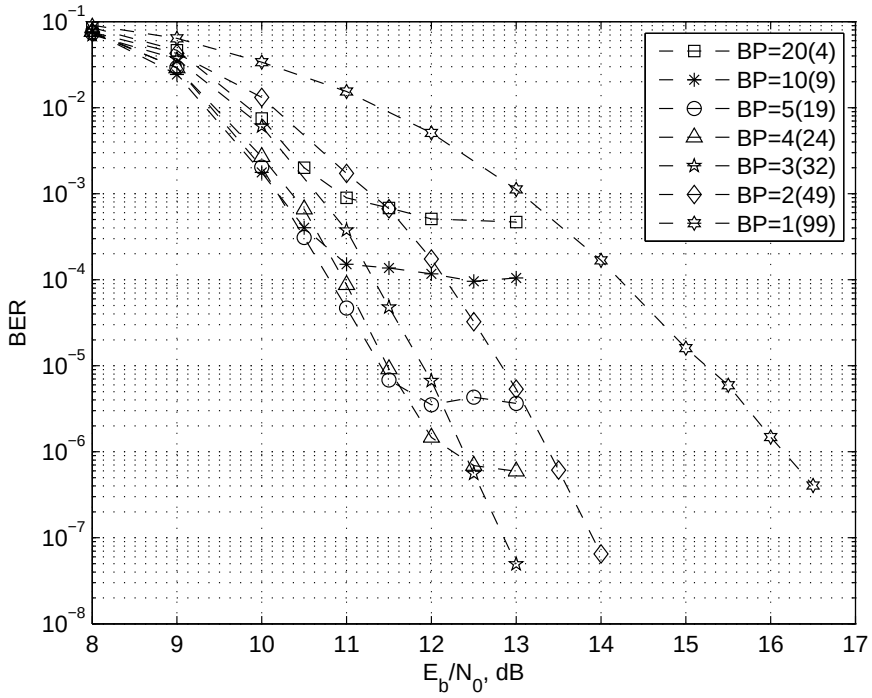


Figure 6.13: BER performance for 16QAM-R2-PIC system using Stopping Criteria II, for $N_{max} = 100$. Throughput is 4.3 bits/symbol period.

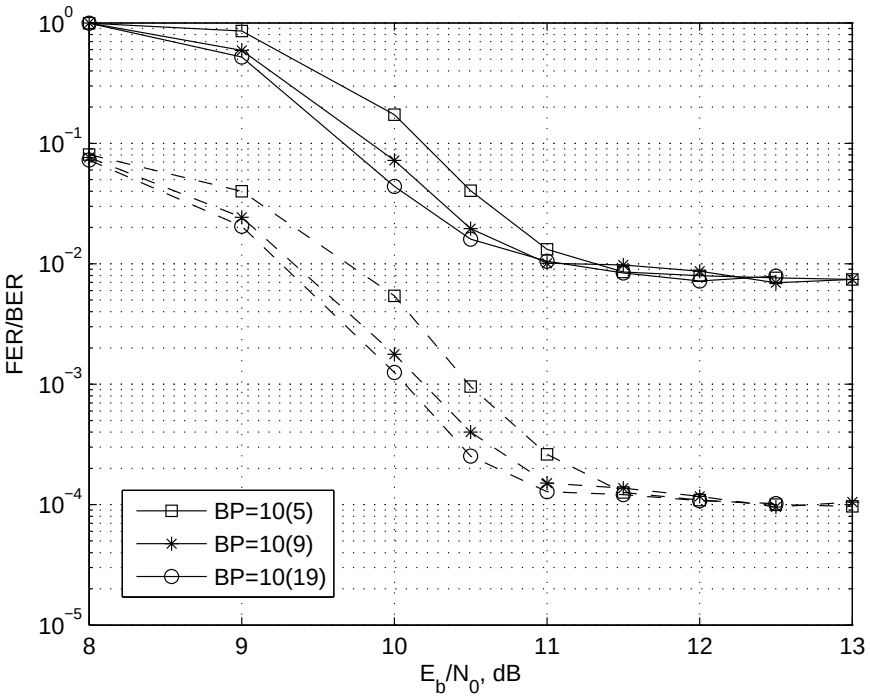


Figure 6.14: BER performance for 16QAM-R2-PIC system using Stopping Criteria II. P_{max} is increased from 5 to 19 for fixed BP_{max}=10. (343,186) LDPC component codes are used giving a throughput of 4.3 bits/symbol period.

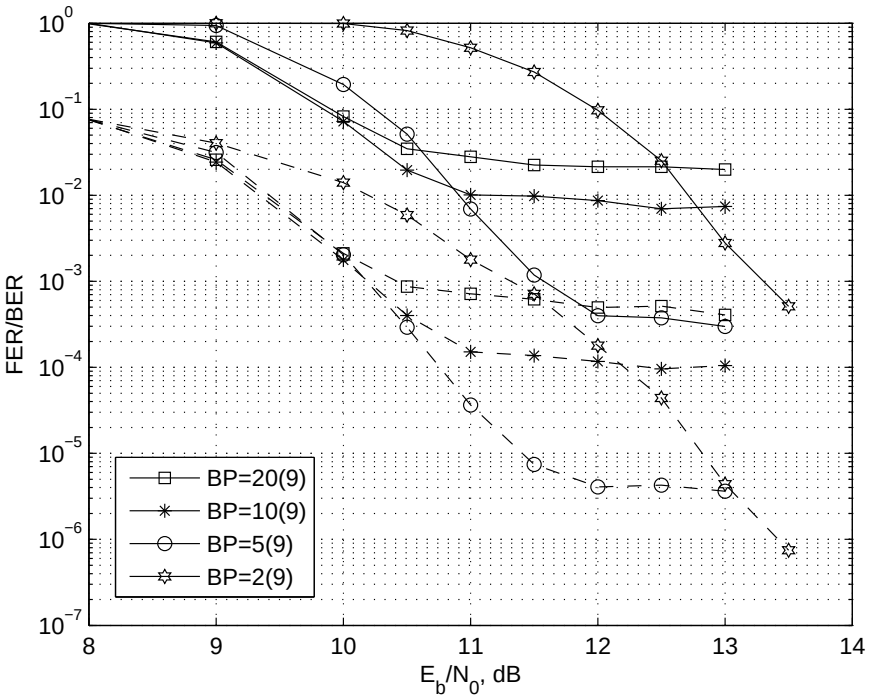


Figure 6.15: BER performance for 16QAM-R2-PIC system using Stopping Criteria II. BP_{max} is increased from 2 to 20 for fixed P_{max}=9. (343,186) LDPC component codes are used giving a throughput of 4.3 bits/symbol period.

5 provides some gain in the BER. No visible gain is observed beyond these values of BP_{max}. Therefore, a sufficient number of BP iterations between PIC updates is required at low SNR. Once this number is reached, additional BP iterations signify wasted processing delays.

At high SNR, however, error floors at BER=5×10^{-4} and BER=1×10^{-4} are observed when BP_{max}=20 and 10, respectively. No error floor is visible down to BER=1×10^{-6} for BP_{max}=2. From Figs. 6.15 and 6.14, it is clear that error floors at high SNR are lowered by increasing the frequency of PIC updates, not the number of PIC updates.

6.7.3 Performance Comparison

Fig. 6.16 compares the effect on performance of using Stopping Criteria I and II. At low SNR, using either stopping criterion provides a small gain over not using any. This is because at low SNR, many bit metrics are incorrect and they are likely to corrupt the correct bit metrics, so it is better to terminate decoding early.

At high SNR, most bit metrics are correct, and the early termination by using stopping criteria denies the chance for these correct bit metrics to influence the incorrect bit metrics. On average, Stopping Criterion II terminates decoding earlier than Stopping Criterion I. Therefore, Stopping Criteria II has a higher error floor at high SNR.

The BP decoding algorithm tries to cause the received vector to converge the received vector to a valid codeword, hopefully the correct codeword. Increasing BP_{max} increases the chances of the BP decoder finding a valid codeword. Since Fig. 6.15 shows that this raises the error floor at high SNR, a significant percentage of these valid codewords must be incorrect codewords. When Stopping Criterion II is used, no additional BP decoding is performed on any sub-label once a valid codeword for that sub-label has been found. Therefore, these incorrect codewords are final and they contribute to the high error floors in the FER and BER. When Stopping Criterion I is used, valid codewords which may be incorrect may be BP decoded again. Termination only occurs when all M valid codewords are found at the same time. This has the effect of reducing the percentage of final incorrect valid codewords which are used to determine the FER and BER. When no stopping criteria are used, we see that no error floors are observed.

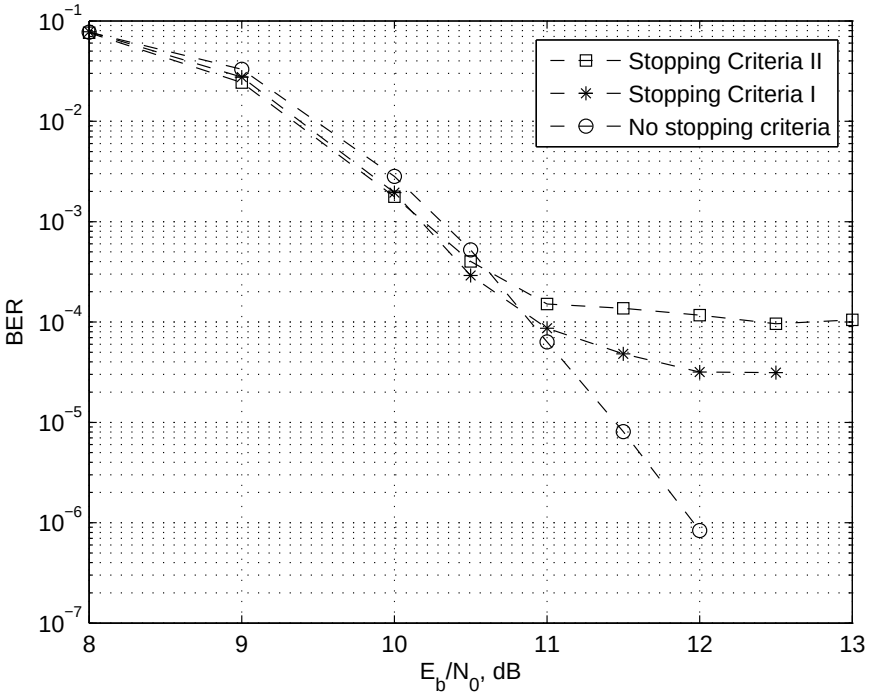


Figure 6.16: BER performance comparison for 16QAM-R2-PIC system using Stopping Criteria I, II and not using any stopping criteria. BP_{max}=10 and P_{max}=9. Throughput is 4.3 bits/s/Hz.

6.8 Summary

In this chapter, we have presented simulation results for the 16QAM-R2-PIC system presented in Chapter 5, using various LDPC component codes. We obtained very good performance at a throughput of around 4.3 bits/symbol period using short length 343 LDPC codes. Even better performance at throughputs of around 5.6 bits/symbol periods were achieved by using longer length 2197 LDPC codes. Due to the structure of BMCM, we showed how by using different codes on sub-labels with different error protection, we can increase throughputs to 6.1 bits/symbol period and still achieve good performance. Throughput can be increased even further by using higher order modulation at the expense of some performance loss. However, we have shown that a better way to increase throughput is to use higher R_{stbc} and/or R_{ldpc}.

When no stopping criteria is used, setting $BP = \sqrt{N_{max}}(\sqrt{N_{max}}-1)$ provides good performance across all SNR. From here the number of PIC updates can be

adjusted to find the best performance. If there were no delay constraints (no N_{max}), both BP_{max} and P_{max} should ideally be made as large as possible.

However, when stopping criteria are introduced to further reduce decoding complexity and delay, increasing BP_{max} and P_{max} provides diminishing gain at low SNR. At high SNR, it is better to decrease BP_{max}, which increases the frequency of PIC updates, to lower the error floors due to convergence to incorrect codewords. Increasing the number of PIC updates on its own (without decreasing BP_{max}) does not appear to lower these floors.

Chapter 7

Practical Considerations

Assumptions enable engineers to break down real-world problems into more manageable components. Research can then be carried out on each component to achieve performance improvements. Once improvements are achieved, the assumptions are removed in order to evaluate how this translates into more realistic scenarios and hence lead to real system performance improvement.

In Chapter 6, all the results presented assumed that perfect channel state information (CSI) was available at the receiver and that the channel faded in an independent quasi-static manner (the QS-Ind channel model). Although these assumptions are also widely adopted throughout literature, they are unrealistic. In Chapter 6, the proposed high throughput BMCM-QOSTBC-PIC system was shown to have excellent performance at low complexity. Here, we come somewhat closer to evaluating the same system in a more practical environment. In Section 7.1, we investigate the effects on performance of channel estimation error. In Section 7.2, we remove the quasi-static fading assumption and observe performance in a time-varying fading channel with different normalized fade rates, $f_D T$. Finally, a simple ad-hoc way of improving performance under different system conditions is presented in Section 7.3.

7.1 Imperfect CSI

In reality, the CSI estimates obtained by a channel estimator are not perfect. The impact of channel estimation error on uncoded orthogonal space-time block codes

(OSTBCs) is investigated in [73, 74, 75]. Here we investigate the impact of channel estimation error on quasi-orthogonal space-time block codes (QOSTBCs) in a bit-mapped coded modulation (BMCM) system using low density parity check (LDPC) component codes. The imperfect CSI model used is presented next, followed by its impact on the performance of the proposed system.

7.1.1 Imperfect CSI model

In [74], channel estimation is performed using pilot symbol assisted modulation (PSAM) together with a sinc interpolator. Pilot symbols are also used in [73] (in a cdma2000 context) and in [75] (in an orthogonal frequency division multiplexing (OFDM) context). Here we are interested in the impact of channel estimation error, regardless of the channel estimator used. Therefore, we represent the estimated CSI[1] at the receiver as

$$\hat{\alpha} = \rho\alpha + \sqrt{1-\rho^2}\epsilon, \qquad 0 \leq \rho \leq 1 \tag{7.1}$$

where α is the actual CSI, ρ is the power correlation coefficient between α^2 and $\hat{\alpha}^2$, and ϵ is an independent zero-mean complex Gaussian random variable with variance 1/2 per dimension. This model is used so that the estimated CSI has the same variance as the actual CSI. We evaluate the amount of estimation noise energy added as $\frac{1-\rho^2}{\rho^2} \times 100\%$ with respect to the actual CSI energy. Using fixed ρ provides the worst-case performance. In [76], it was shown that ρ approaches 1 with increasing SNR. When $\rho = 1$, we have $\hat{\alpha} = \alpha$, meaning perfect CSI is available.

7.1.2 Performance Using Imperfect CSI

Fig. 7.1 compares the effects of CSI error on performance of the 16QAM-R1-JD and QPSK-R2-PIC systems. Both schemes use (343,186) LDPC component codes. When perfect CSI is available ($\rho = 1$), the QPSK-R2-PIC system outperforms the 16QAM-R1-JD system by 5dB at a BER of 10^{-4}. This gain increases as the amount of CSI estimation error increases (decreasing ρ).

We now look at the sensitivity of both systems to CSI errors. Compared against the performance with perfect CSI, the coded 16QAM-R1-JD system loses

[1]Actual estimation of CSI at the receiver is beyond the scope of this thesis. Only modelling of a random estimation error is considered.

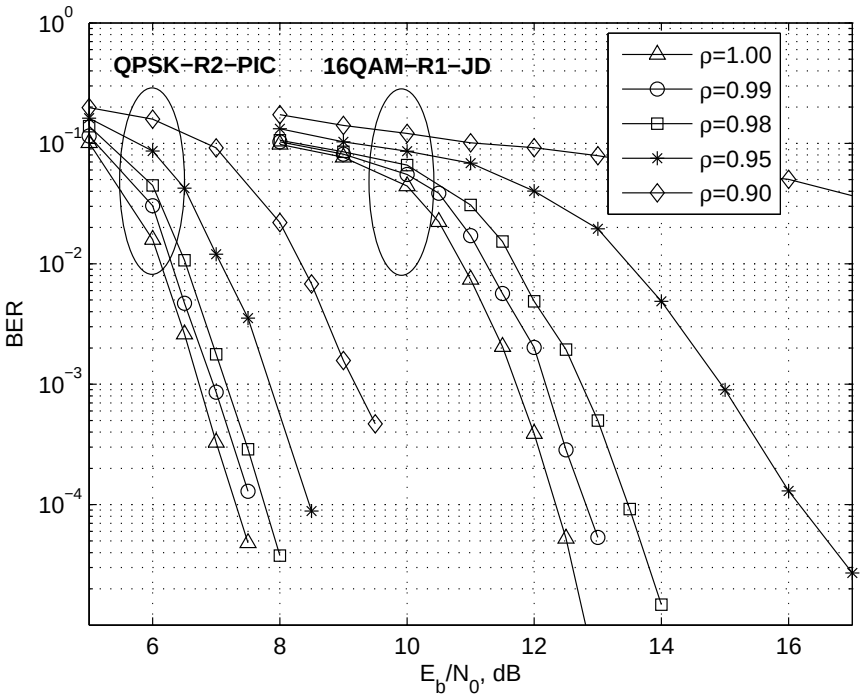


Figure 7.1: BER performance for the 16QAM-R1-JD and QPSK-R2-PIC BMCM systems when ρ=1, 0.99, 0.98, 0.95 and 0.90. Both schemes use the (343,186) LDPC component codes. Throughput is 2 bits/symbol period.

1.2dB and 3.9dB at BER=10^{-4}, for $\rho = 0.98$ and $\rho = 0.95$ respectively. This is equivalent to an average estimation error energy of about 4% and 11% of the actual CSI, respectively. The coded QPSK-R2-PIC system loses only 0.5dB and 1.1dB in the same scenario. Simulations also found that the coded 16QAM-R1-PIC system loses 0.7dB and 2.1dB compared to perfect CSI. The sensitivities of all three systems to CSI errors are summarized in Table 7.1. All these systems have a throughput of 2 bits/symbol period.

Higher order modulation schemes have a smaller minimum squared Euclidean distance between constellation points, and hence are more sensitive to channel estimation errors. Comparing the performance of QPSK-R2-PIC and 16QAM-R1-PIC, we find that the SNR gain due to the extra estimation noise protection offered by QPSK exceeds the SNR loss due to the additional co-channel interference introduced by the Rate 2 QOSTBC. This demonstrates the effectiveness of using higher rate STBCs instead of using higher order modulations in order to achieve the same

System	$\rho = 0.98$	$\rho = 0.95$
coded QPSK-R2-PIC	0.5	1.1
coded 16QAM-R1-PIC	0.7	2.1
coded 16QAM-R1-JD	1.2	3.9

Table 7.1

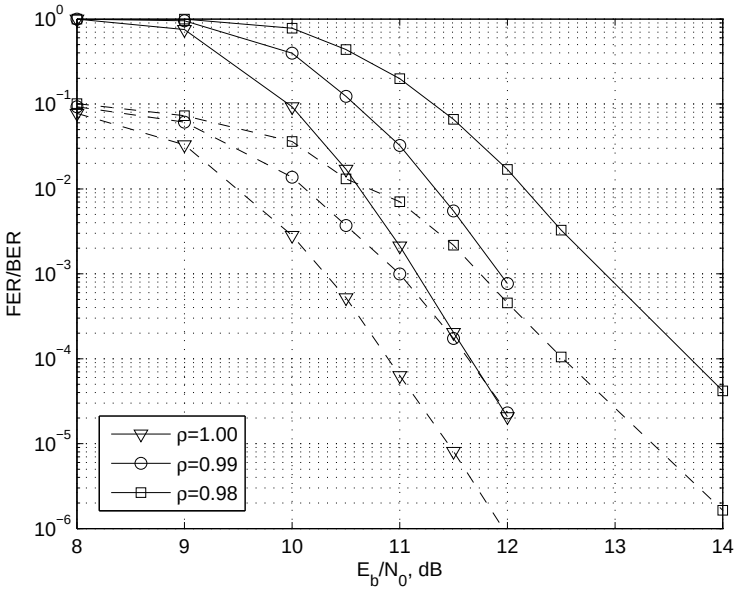
Sensitivities of different BMCM-QOSTBC systems to channel estimation errors, measured as SNR losses (in dB) with respect to perfect CSI ($\rho = 1$) at BER=10^{-4}. Throughput is 2 bits/symbol period.

system throughput. This also suggests that it is better to increase throughput by using higher rate STBCs rather than higher order modulation.

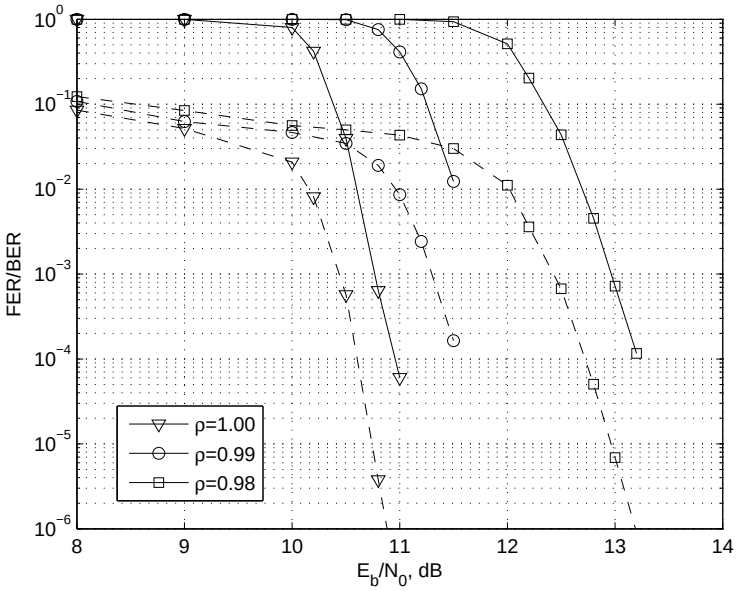
Comparing 16QAM-R1-JD and 16QAM-R1-PIC, we find that JD is more sensitive to imperfect CSI than PIC. The JD scheme also suffers a diversity loss in the presence of estimation errors. The iterative nature of PIC detection together with BP decoding improves the robustness of our system to channel estimation errors, compared to one-off JD detection followed by BP decoding.

It is obvious from Fig. 7.1 that the QPSK-R2-PIC system is very robust to channel estimation errors. It only loses about 2.6dB compared to the perfect CSI case when $\rho = 0.90$, which is equivalent to an average estimation error energy of approximately 23%. In addition, no diversity loss can be seen.

Fig. 7.2 show the effects of CSI estimation error on the FER and BER performances for 16QAM-R2-PIC using short (343,186) and long (2197,1544) LDPC component codes. Each frame corresponds to the length of the LDPC code, which spans 172 and 1100 time slots, respectively, for the short and long LDPC codes. When $\rho = 0.98$, the BER for the short LDPC codes reaches 10^{-4} at 12.5dB, an SNR loss of 1.6dB from the perfect CSI case. When longer LDPC component codes are used, a loss of about 2.1dB is observed. The long LDPC codes enable the system to achieve a throughput of 5.6 bits/symbol period. The BER curves for the long LDPC codes are falling at around 7 decades/dB (for $\rho = 1$) and 4 decades/dB ($\rho = 0.98$). No error floors are observed down to FER=10^{-4} and BER=10^{-6}. Throughputs can be further increased by increasing R_{ldpc}, but this may lead to error floors being observed.



(a) Using (343,186) LDPC component codes (4.3 bits/symbol period).



(b) Using (2197,1544) LDPC component codes (5.6 bits/symbol period).

Figure 7.2: FER (solid lines) and BER (dashed lines) performances for the 16QAM-R2-PIC system. Perfect (ρ=1) and imperfect (ρ=0.99 and 0.98) CSI are considered. All LDPC codes use BP=10(9).

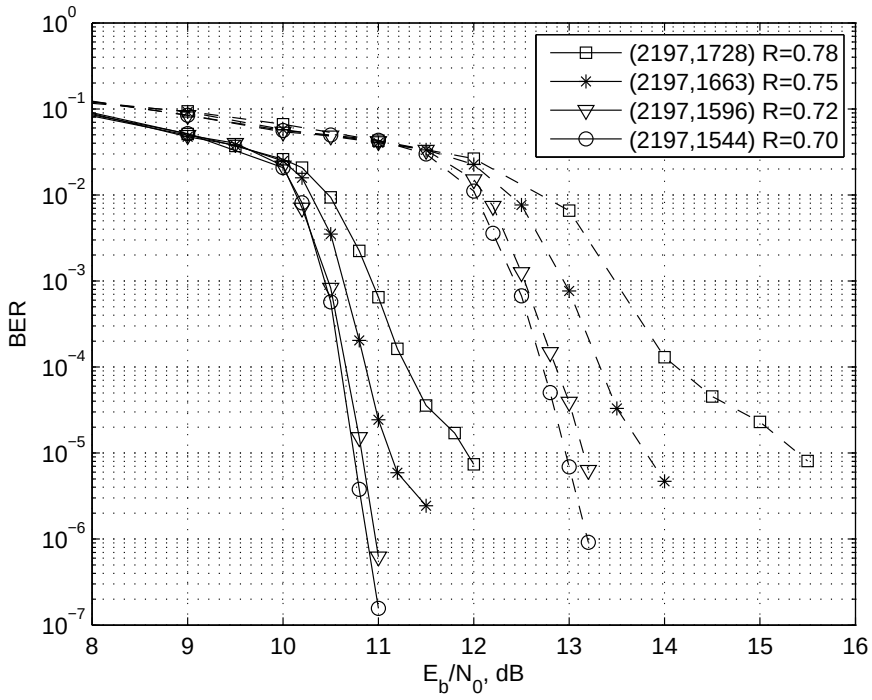


Figure 7.3: BER performance for the 16QAM-R2-PIC system using the same length 2197 LDPC component code on all labels ABCD, for ρ=1 (solid lines) and ρ=0.98 (dashed lines). LDPC code rate varies from 0.70 to 0.78, giving throughputs of 5.6 to 6.3 bits/symbol period.

We now focus on the long (length 2197) LDPC component codes. Fig. 7.3 shows the effects of CSI estimation error on the BER performance of these codes in 16QAM-R2-PIC systems with different R_{ldpc}. Compared to the perfect CSI case, the BER performances at BER=10^{-5} for $\rho = 0.98$ suffer SNR losses of 2.2dB, 2.3dB, 2.7dB and 3.5dB for $R_{ldpc} = 0.70, 0.72, 0.75$ and 0.78, respectively. The diversity losses (determined by the slope differences of BER curves) also increase in the same order. Therefore, system sensitivity to CSI estimation errors increases with R_{ldpc}. At $\rho = 0.98$, the (2197,1544) LDPC component codes achieve excellent performance with the BER falling at around 4.3 decades/dB and no error floor is observed down to BER=10^{-6}. There is thus a small diversity loss compared to the $\rho = 1$ case, as seen in Fig. 7.3.

In Section 6.6, the benefits of using codes with unequal error correction capability on sub-labels with different error protection levels are discussed. We identified System I as the 16QAM-R2-PIC system which uses the same (2197,1663)

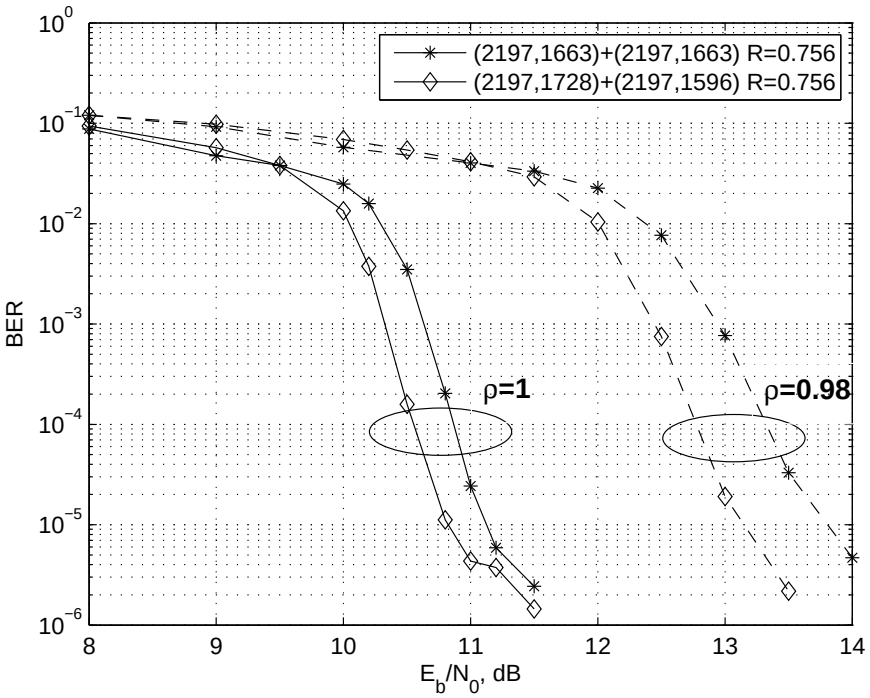


Figure 7.4: BER performance for the 16QAM-R2-PIC system using the (2197,1663) LDPC component code on all labels ABCD (System I), and using (2197,1728) LDPC component code on AB and (2197,1663) LDPC component code on CD, for ρ=1 and ρ=0.98. Throughput is 6.0 bits/symbol period.

LDPC component codes on all sub-labels ABCD, and System II as the 16QAM-R2-PIC system which uses (2197,1728) LDPC component codes on sub-labels AB and (2197,1596) LDPC component codes on sub-labels CD, giving $R_{ldpc} = 0.75$ in both systems. Fig. 6.10 showed how System II outperforms System I when perfect CSI is available. Here, we consider imperfect CSI with $\rho = 0.98$.

Fig. 7.4 shows that System II outperforms System I even in the presence of CSI estimation errors. At $\rho = 0.98$, System II outperforms System I by 0.65dB, compared to only 0.30dB when $\rho = 1$. We have seen from Fig. 6.10 that the overall BER is dominated by sub-labels CD, for both systems. System II uses a lower rate LDPC component code on CD compared to that in System I. From Fig. 7.3, we know that lower rate LDPC codes are more robust to CSI errors. Hence, this explains why System II is more robust to CSI estimation errors.

7.2 Time-varying Fading Channels

The quasi-static fading assumption used in the QS-Ind and QS-fDT channels allows simple linear processing to be used for detection. However, this is not realistic in practice. The performance of uncoded OSTBCs in time-varying channels are investigated in [77, 78]. Here, we investigate the performance of the LDPC coded QOSTBCs in time-varying fading channels. Fig. 7.5 shows the effects of removing the quasi-static assumption during transmission, which transforms the QS-fDT channel into the NoQS-fDT channel. We consider f_DT in the range of 0.1 to 0.0001. We use QPSK and the Rate 2 QOSTBC together with two (343,186) LDPC component codes, giving a throughput of 2 bits/symbol period. For the NoQS-fDT channel, a 1.6dB loss over the QS-fDT channel is observed when the fading becomes very fast (f_DT=0.1). The SNR loss arises because the NoQS-fDT channel changes significantly over the L time slots of the STBC when f_DT=0.1, contributing to large CSI

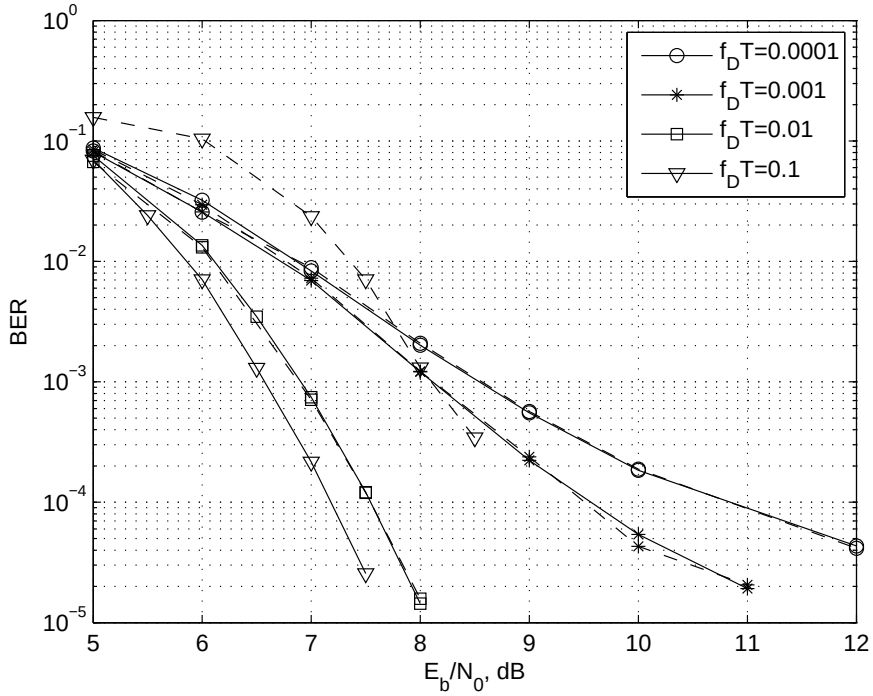


Figure 7.5: BER performance for the QPSK-R2-PIC system in the QS-fDT channel (solid lines) and the NoQS-fDT channel (dashed lines), for normalized fade rates of f_DT = 0.1 - 0.0001. Both schemes use the (343,186) LDPC component codes with BP=10(9). Perfect CSI is assumed. Throughput is 2 bits/symbol period.

estimation errors when the quasi-static assumption is used during linear processing. Removing the quasi-static assumption has almost no effect on performance for the more realistic f_DT values of 0.01 (fast fading) to 0.0001 (slow fading).

On the other hand, as the f_DT values decrease from 0.01 to 0.0001 in both the QS-fDT and NoQS-fDT channels, the channel variation over a LDPC codeword decreases significantly. This results in a loss of diversity gain as the fading becomes slower.

An interesting point to note is the reversal in performance trend between uncoded OSTBCs and coded QOSTBCs. In uncoded OSTBC using linear processing, performance degrades as the fading becomes faster [77, 78], due to increasing CSI estimation error. This results in irreducible error floors at high SNR, which rise as the fading becomes faster. PIC is applied on OSTBCs to eliminate these floors in [78]. In coded QOSTBCs, performance generally degrades as the fading becomes slower, due to the lack of channel variation seen by the error-correcting codes. The exception occurs when the fading becomes impossibly fast $f_DT = 0.1$, at which point the effect of CSI estimation error takes over.

7.2.1 Effect of increasing P_{max} and BP_{max}

The simulation results shown in Fig. 7.5 for the QPSK-R2-PIC system use BP=10(9) in the detection/decoding process. We now investigate the effect of increasing P_{max} from 9 to 39 in the same system. In order to make our simulations resemble more practical systems, we also add a CSI estimation error of 4%($\rho = 0.99$). Fig. 7.6 shows that increasing P_{max} from 9 to 39 (while maintaining $BP_{max} = 10$) provides negligible gains in the fast fading scenario ($f_DT = 0.01$). However, when the fading is slow ($f_DT = 0.0001$), increasing P_{max} from 9 to 19 provides significant diversity and coding gains. When P_{max} is further increased to 39, significant gains are also obtained at high SNR (>10dB) but not at low SNR. This suggests that performance gains in slow fading are only provided up to a certain value of P_{max}, however, this value increases with SNR. The FER performances exhibit similar trends.

Fig. 7.7 shows the effect on performance of increasing BP_{max} from 5 to 20 (while maintaining $P_{max} = 19$). In both fast and slow fading, increasing BP_{max} provides minimal coding gains and no diversity gains. The FER performances also

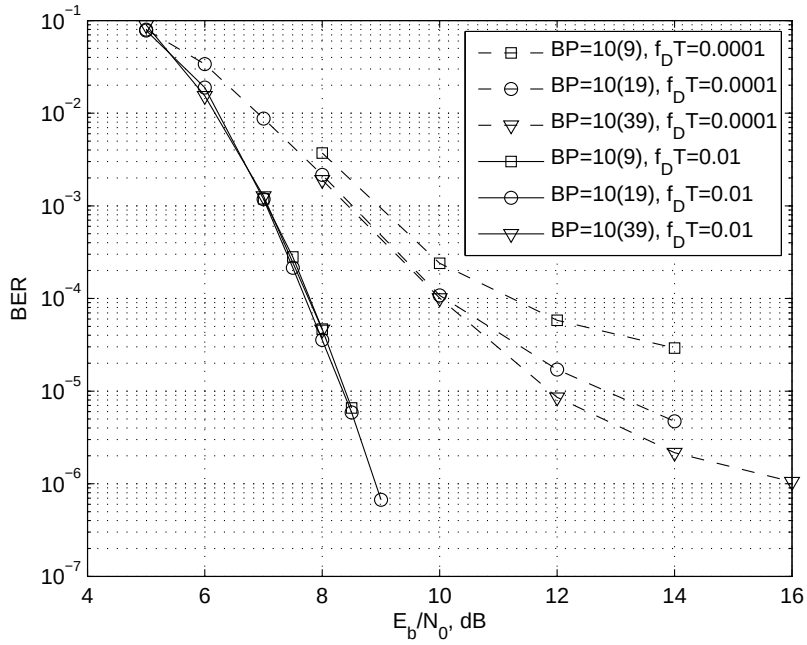


Figure 7.6: BER performance for the QPSK-R2-PIC system in the NoQS-fDT channel, when P_{max} is increased from 9 to 39 for $f_DT = 0.0001$ and 0.01. The (343,186) LDPC component codes are used and imperfect CSI with $\rho = 0.99$ is assumed. Throughput is 2 bits/symbol period.

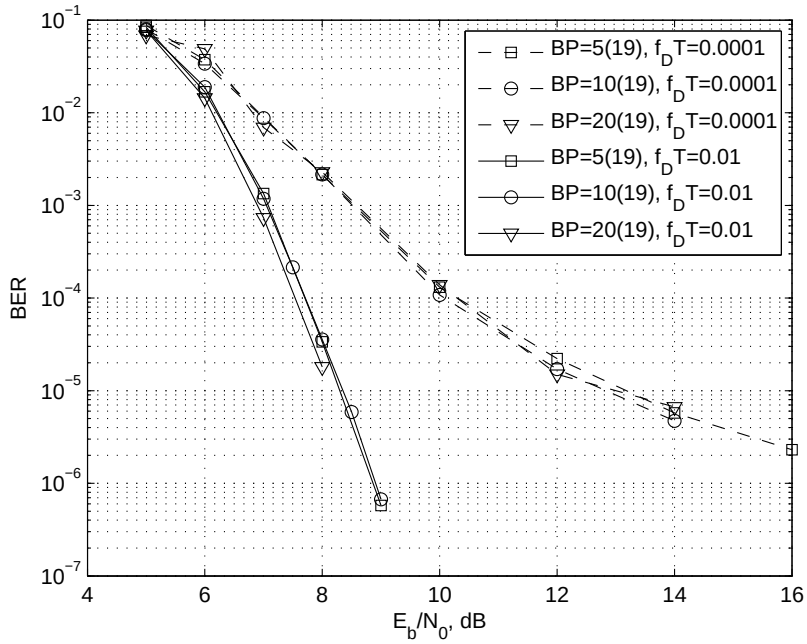


Figure 7.7: BER performance for the QPSK-R2-PIC system in the NoQS-fDT channel, when BP_{max} is increased from 5 to 20 for $f_DT = 0.0001$ and 0.01. The (343,186) LDPC component codes are used and imperfect CSI with $\rho = 0.99$ is assumed. Throughput is 2 bits/symbol period.

exhibit similar trends, with the exception of the $BP_{max} = 5$ case which suffers a 0.5dB loss across all SNR in both fast and slow fading. This is because when BP_{max} is too small, each frame (which spans the codeword length) has a small chance of converging to a valid codeword, thereby resulting in a frame error.

7.2.2 Performance comparison

Fig. 7.8 compares the coded FER performance of QPSK-R2-PIC and 16QAM-R1-JD systems using (343,186) LDPC component codes. Here, we use BP=10(19) for the PIC system, giving $N_{max} = 200$. The JD system also uses $N_{max} = 200$. When the fading is slow ($f_D T = 0.0001$), QPSK-R2-PIC outperforms 16QAM-R1-JD by about 5.5dB at $FER = 10^{-3}$. However, the QPSK-R2-PIC system suffers a diversity loss at high SNR. This loss in diversity can be eliminated by employing more than 19 PIC updates (and hence larger N_{max}). When the fading is fast ($f_D T = 0.01$),

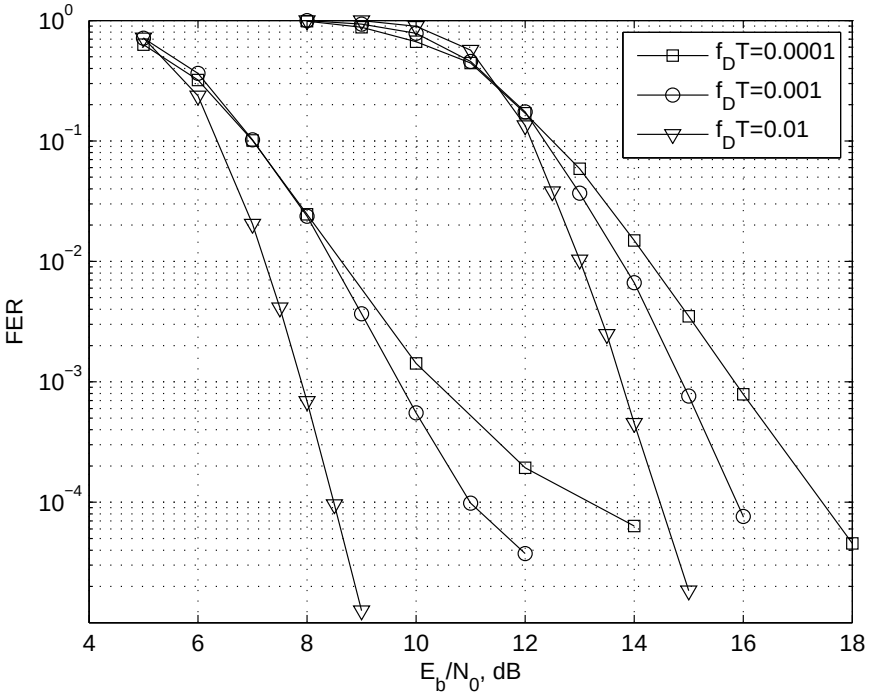
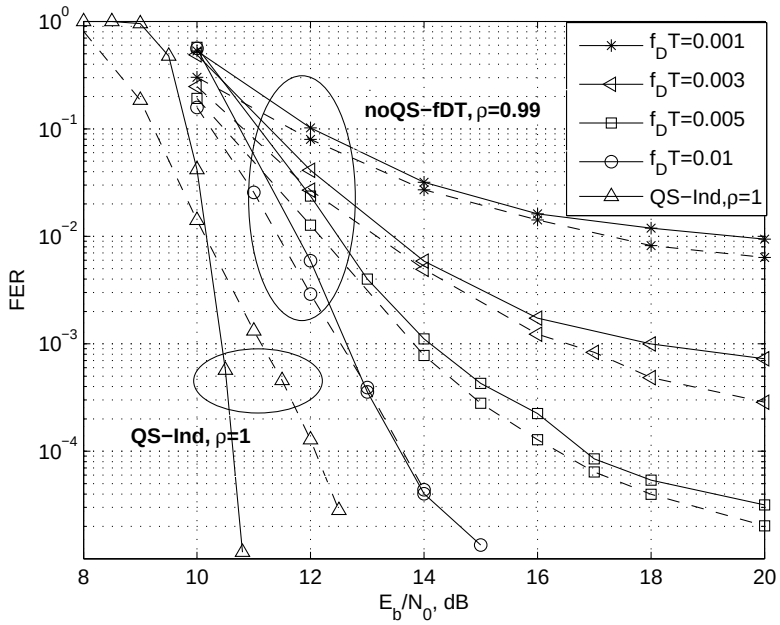
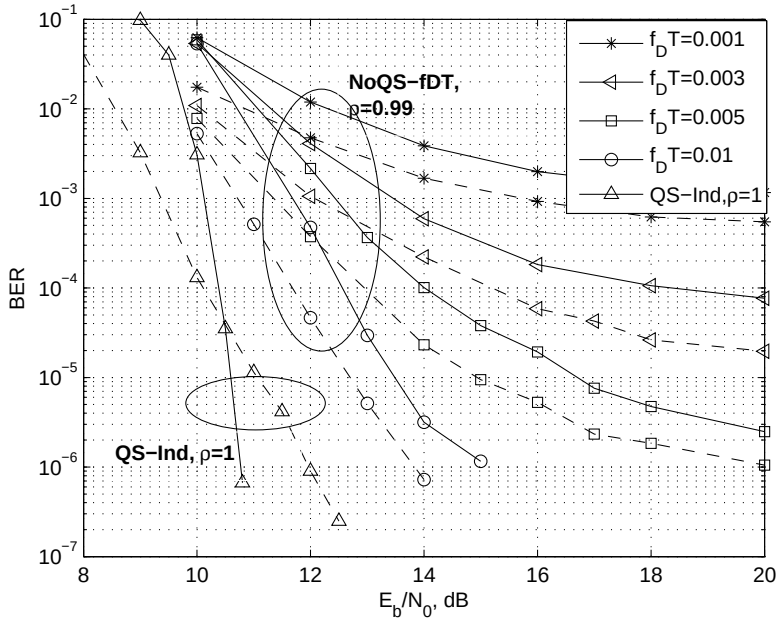


Figure 7.8: FER performances for the QPSK-R2-PIC and 16QAM-R1-JD systems in the NoQS-fDT channel, for normalized fade rates of $f_D T = 0.01$ - 0.0001. Both systems use the (343,186) LDPC component codes. A CSI estimation error with $\rho = 0.99$ is assumed. Throughput is 2 bits/symbol period.



(a) FER performance.



(b) BER performance.

Figure 7.9: Performance comparison for 16QAM-R2-PIC BICM (solid lines) and BMCM (dashed lines) systems in the QS-Ind channel with $\rho = 1$ and in the NoQS-fDT channel with $\rho = 0.99$, for normalized fade rates of 0.01 - 0.001. BP=10(9) is used for both systems. Throughput is 4.3 bits/symbol period.

QPSK-R2-PIC outperforms 16QAM-R1-JD at the same FER by approximately 6dB, without any signs of an error floor. Similar trends are exhibited in the BER case, for both systems.

In Section B we compared the structure of bit-interleaved coded modulation (BICM) to BMCM. In Fig. 7.9, we compare the performance for both systems using 16-QAM and the Rate 2 QOSTBC. The BMCM system uses two (343,216) LDPC codes on sub-labels AB and two (343,156) LDPC codes on sub-labels CD to match the unequal error protection of 16-QAM. This gives an overall throughput of approximately 4.3 bits/symbol period. It spans 172 time slots. The BICM system uses a single (1331,730) LDPC code and spans 167 time slots. Both systems have approximately the same system throughput.

In the QS-Ind channel with perfect CSI, BMCM outperforms BICM down to $FER = 1 \times 10^{-2}$ and $BER = 4 \times 10^{-5}$. Although not shown here, the crossover points with BICM are $FER = 2 \times 10^{-1}$ and $BER = 3 \times 10^{-3}$ if four (343,186) LDPC component codes were used in the BMCM system. This demonstrates the advantage of the BMCM structure compared to BICM in optimizing performance, for modulations where the sub-labels have unequal error protection (typical in higher order modulation). BICM outperforms BMCM by $1.5dB$ at $FER = 10^{-4}$, due to the higher minimum distance of the longer LDPC code.

However, BMCM achieves comparable FER performance to BICM in a more realistic NoQS-fDT channel, for normalized $f_D T$ values of 0.01-0.001 and a 2% CSI estimation error. BMCM also achieves FER error floors approximately 1.6 to 3 times lower than BICM. The BER error floors for BMCM are approximately 2 to 5 times lower than BICM at high SNR. The lower floors on BMCM is due to the shorter codes used. In slower fading, there are so many errors across each codeword that it is better to not use error-correcting codes. This is analogous to the low SNR region when uncoded systems outperform coded systems. However, it was shown in Fig. 6.2 that PIC only works with coding. Therefore, a good compromise is to use short codes, which is natural in the BMCM structure. The error floors rise as $f_D T$ decreases. As with the QPSK case in Fig. 7.6, these floors can be lowered by using more PIC updates.

While BICM benefits from using longer codewords in a QS-Ind channel due to

the higher minimum distance of the code, the same codeword becomes a burden in realistic channel conditions. As discussed in Section B, the BMCM structure shortens processing delay by a factor of 4 compared to BICM. In addition, unlike BICM, it does not suffer delays due to interleaving/de-interleaving. Therefore, BICM does not meet our design criterion in Section 1.3.

7.2.3 Performance improvement in slow fading

In Fig. 7.8, the QPSK-R2-PIC system was shown to suffer a diversity loss in slow fading as SNR increases. Its FER is falling[2] at 0.24 decades/dB using BP=10(19), compared to 0.60 decades/dB for the 16QAM-R1-JD system. In Fig. 7.6, it was shown that this diversity loss can be recovered by employing more PIC updates. By setting BP=10(39), the FER for the QPSK-R2-PIC system falls at 0.37 decades/dB. However, this increases N_{max} from 200 to 400, which increases delay. Here, we investigate the effect of fixing N_{max} and varying the frequency of PIC updates.

Fig. 7.10 shows the FER and BER performances for the QPSK-R2-PIC system in slow fading ($f_D T = 0.0001$). N_{max} is fixed at 100. By increasing the frequency of PIC updates from BP=10(9) to BP=5(19), we get a BER performance gain at all the SNR shown, and an FER performance gain at high SNR (>10dB). However, at low SNR, the FER suffers a small performance loss.

As discussed in Section 7.2.1, increasing P_{max} at high SNR in slow fading provides diminishing performance gains. However, this decreases the allowable BP_{max} which degrades the FER performance across all SNR. Therefore, as we increase the frequency of PIC updates from BP=10(9) to BP=5(19), the FER performance gain obtained at high SNR due to increasing P_{max} outweighs the performance loss due to decreasing BP_{max}. However at low SNR, the FER performance gain from increasing P_{max} decreases while the FER performance loss from decreasing BP_{max} remains constant, resulting in an overall FER performance loss. Decreasing BP_{max} reduces the chance of the BP decoder converging. This results in increased FER but does not necessarily degrade the BER. Therefore, the BER performance gain due to increasing P_{max} outweighs the BER performance loss due to decreasing BP_{max} at all SNR shown in Fig. 7.10.

[2]Measured as the slope around FER=10^{-4}.

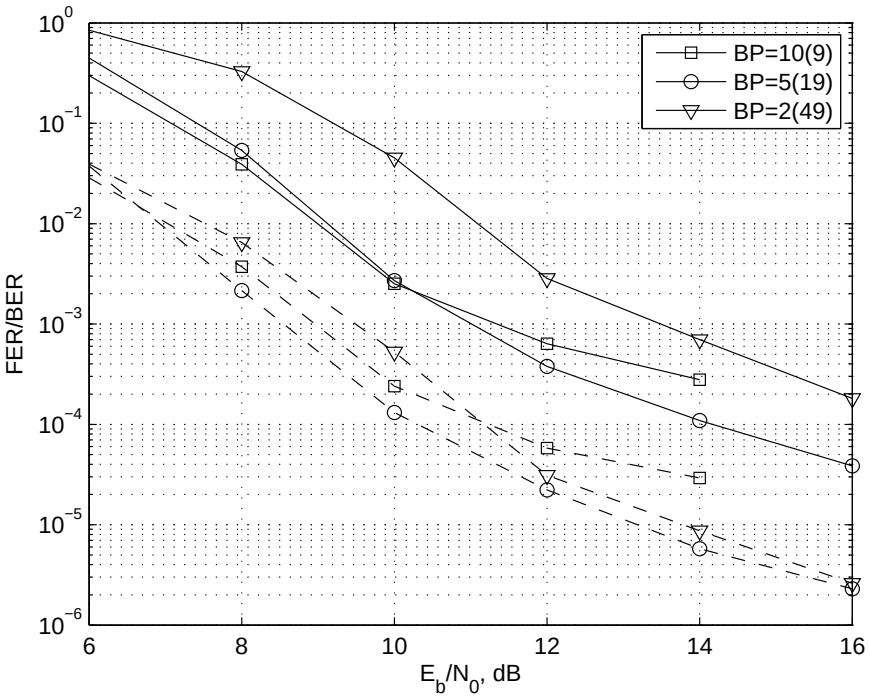


Figure 7.10: FER and BER performances for the QPSK-R2-PIC systems in the NoQS-fDT channel, for $f_D T = 0.0001$ using (343,186) LDPC component codes. BP=10(9), BP=5(19) and BP=2(49) are used. A CSI estimation error with $\rho = 0.99$ is assumed. Throughput is 2 bits/symbol period.

However, when the frequency of PIC updates is increased from BP=5(19) to BP=2(49), the probability of BP convergence is reduced significantly. This results in approximately a 2dB FER performance loss across all SNR shown in Fig. 7.10. Both the FER and BER performances using BP=2(49) fall faster than those using BP=5(19). In addition, the performance crossover points between the BP=5(19)/BP=2(49) pair[3] occur at a higher SNR (over 16dB) compared to those between the BP=10(9)/BP=5(19) pair (10.2dB for FER and 6.6dB for BER). This is consistent with the observation in Fig. 7.6 which shows how performance gain due to increasing P_{max} diminishes with increasing SNR.

It is clear that increasing P_{max} as SNR increases can improve the performance of the QPSK-R2-PIC system in slow fading. However, when N_{max} is fixed, this decreases BP_{max}, which degrades performance due to the non-convergence of

[3]Interpolation is required.

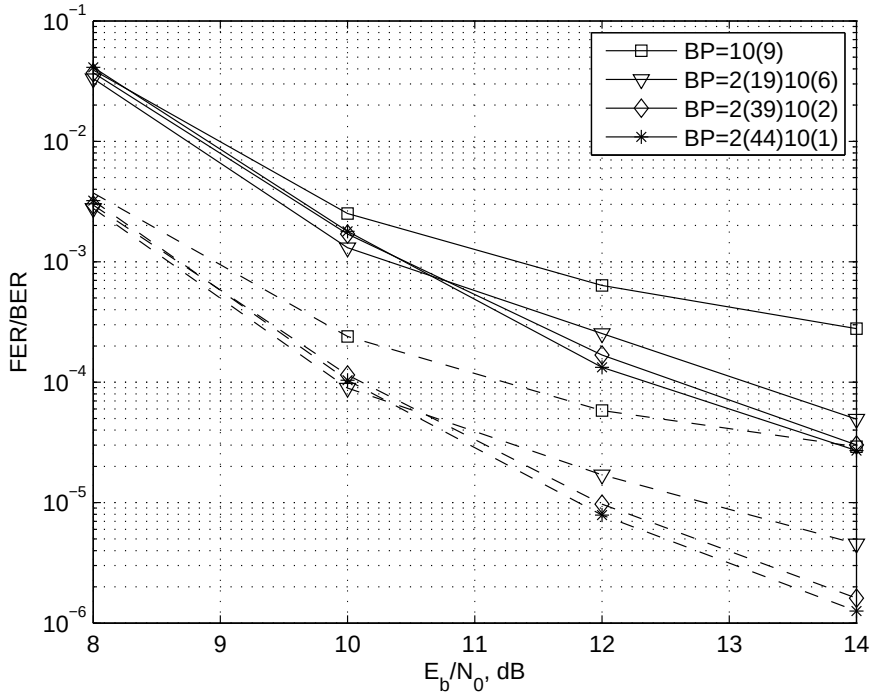


Figure 7.11: FER and BER performances for the QPSK-R2-PIC systems in the NoQS-fDT channel, for $f_D T = 0.0001$ using (343,186) LDPC component codes. BP=10(9), BP=(2)(19)(10)(6), BP=(2)(39)(10)(2) and BP=(2)(44)(10)(1) are considered. A CSI estimation error with $\rho = 0.99$ is assumed. Throughput is 2 bits/symbol period.

the BP decoders. Here, we propose dividing N_{max} into two parts - $N_{max,a}$ followed by $N_{max,b}$, where $N_{max} = N_{max,a} + N_{max,b}$. This is denoted by the notation $BP = (BP_{max,a})(P_{max,a})(BP_{max,b})(P_{max,b})$ where $N_{max,a} = BP_{max,a} + BP_{max,a} \times P_{max,a}$ and $N_{max,b} = BP_{max,b} \times P_{max,b}$. Frequent PIC updates are performed during $N_{max,a}$, while infrequent PIC updates are performed during $N_{max,b}$. Fig. 7.11 illustrates the FER and BER performances for the QPSK-R2-PIC system in slow fading using different combinations of $N_{max,a}$ and $N_{max,b}$. BP_{max} is set to 2 in $N_{max,a}$ (frequent PIC updates) and 10 in $N_{max,b}$ (infrequent PIC updates). As we increase the number of frequent PIC updates from 0 to 44, both the FER and BER performances also improve. The diversity achieved by the system also increases when the number of frequent PIC updates, $P_{max,a}$, is increased. We know that in the limit when $P_{max,a} = P_{max}$, performance degrades due to the non-convergence of the BP decoders. However, the excellent performance achieved using BP=(2)(44)(10)(1)

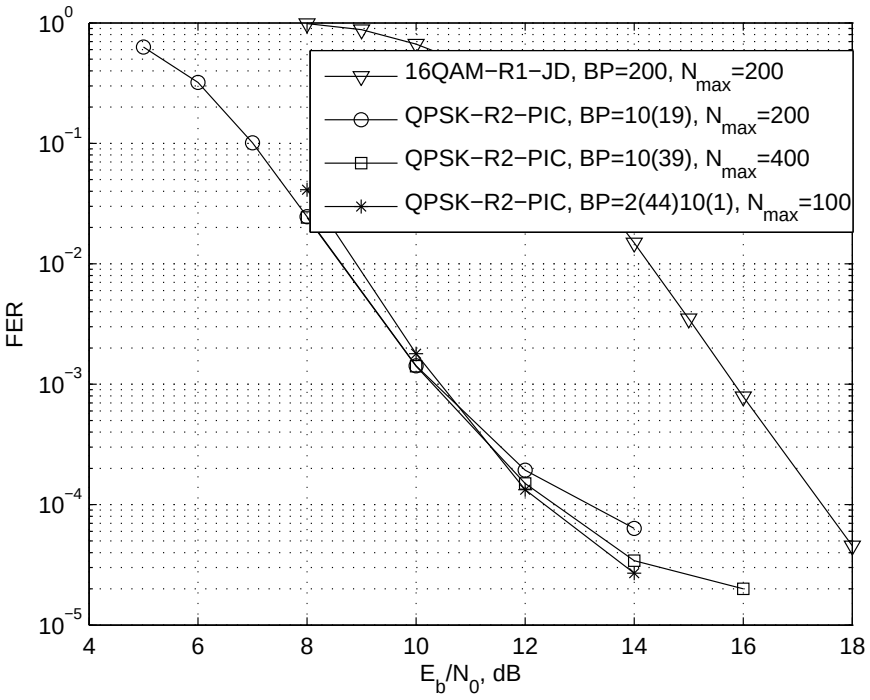


Figure 7.12: FER performances for the QPSK-R2-PIC and 16QAM-R1-JD systems in the NoQS-fDT channel, for $f_D T = 0.0001$. Both systems use the (343,186) LDPC component codes. A CSI estimation error with $\rho = 0.99$ is assumed. Throughput is 2 bits/symbol period.

shows that most of the convergence occurs within $BP_{max,b} = 10$ iterations and we only need to perform this once for the short (343,186) LDPC component codes at the very end of the joint PIC-BP detection/decoding process. By breaking a fixed N_{max} into $N_{max,a}$ and $N_{max,b}$, we obtain the benefits of frequent PIC updates without sacrificing performance due to non-convergence of the BP decoders.

Fig. 7.12 shows how the diversity loss of the QPSK-R2-PIC system compared to the 16QAM-R1-JD system at high SNR in slow fading channels can be minimized by increasing P_{max} from 19 to 39. However, this increases N_{max} from 200 to 400. A more effective way is to combine the use of frequent and infrequent PIC updates, using BP=(2)(44)(10)(1) (giving $N_{max} = 100$). This outperforms BP=10(39) (giving $N_{max} = 400$) while reducing N_{max} by 75%.

7.3 Decoding Metric Improvement

In Chapter 5, we showed how the soft input bit metrics to the BP decoders are traditionally defined following [22] as

$$\lambda_i^m \approx \frac{1}{2\sigma^2}\left[\min_{\beta\in\chi_i^{m,(0)}}||\hat{\boldsymbol{s}}_i - K\gamma||^2 - \min_{\beta\in\chi_i^{m,(1)}}||\hat{\boldsymbol{s}}_i - K\beta||^2\right] \tag{7.2}$$

where σ^2 is the variance of co-channel interference and receiver noise. In [71], it was shown that performance at high SNR can be improved by removing the noise variance term from equation (7.2). In addition, further improvement [71] can be achieved by generalizing the Euclidean L_2-norm to an L_p-norm and then reducing p from 2 to 1.9. In this thesis, we have adopted the removal of the noise variance. Furthermore, we have reduced p to 1 to obtain the L_1-norm. This allows us to simplify the calculation of bit metrics to

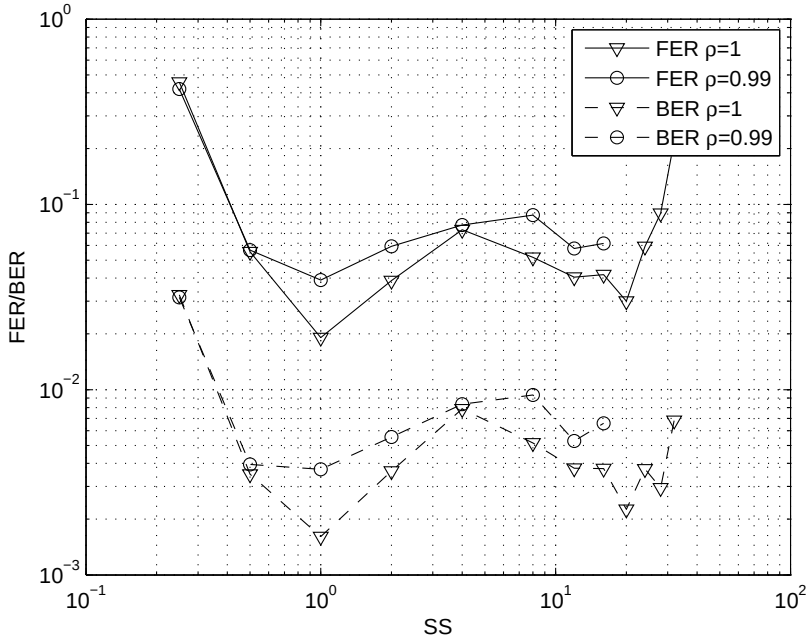
$$\lambda_i^m \approx \min_{\gamma\in\chi_i^{m,(0)}}||\hat{\boldsymbol{s}}_i - K\gamma||^1 - \min_{\beta\in\chi_i^{m,(1)}}||\hat{\boldsymbol{s}}_i - K\beta||^1. \tag{7.3}$$

Both these steps have the effect of scaling the bit metrics. This motivated us to investigate the effect of scaling on bit metrics. To that end, we re-write equation (5.13) as

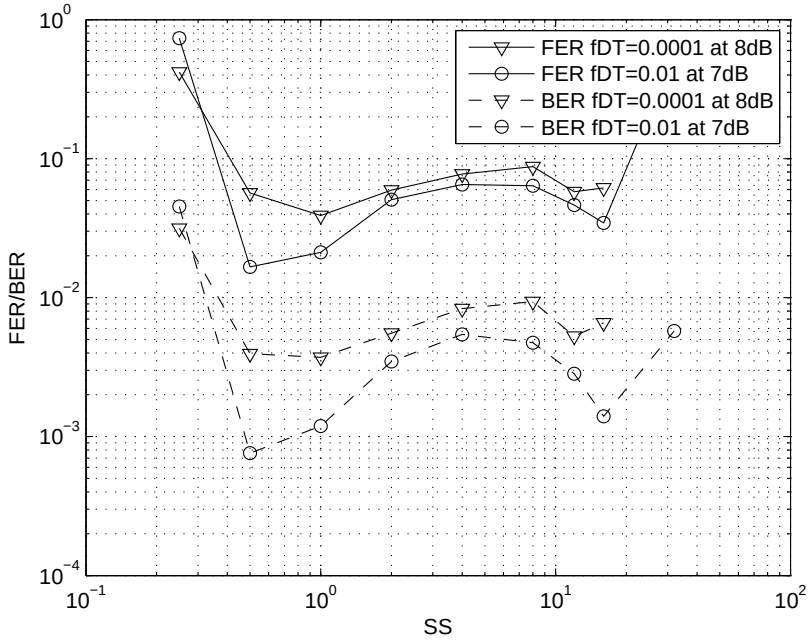
$$\lambda_i^m \approx SS\left[\min_{\gamma\in\chi_i^{m,(0)}}||\hat{\boldsymbol{s}}_i - K\gamma||^1 - \min_{\beta\in\chi_i^{m,(1)}}||\hat{\boldsymbol{s}}_i - K\beta||^1\right] \tag{7.4}$$

where SS is the scaling factor. Unlike the noise variance in equation (7.2), SS is independent of SNR.

Fig. 7.13 shows the effect of SS on BER performance for the QPSK-R2-PIC system. FER and BER performances of this system in NoQS-fDT channels are shown at 8dB for $f_DT = 0.0001$ and 7dB for $f_DT = 0.01$, using SS values between 0.25 and 32. In Fig. 7.13a, we observe that using $SS = 1$ in the slow fading channel provides the best FER and BER performances, for both perfect and imperfect CSI with $\rho = 0.99$. This is encouraging as $SS = 1$ corresponds to the bit metric we decided to use throughout this thesis. However, Fig. 7.13b shows that in fast fading, using $SS = 0.5$ provides slightly better performance.



(a) $f_DT = 0.0001$ with ρ=1 and 0.99.



(b) f_DT=0.0001 and 0.01 with $\rho = 0.99$.

Figure 7.13: FER and BER performances for the QPSK-R2-PIC system using (343,186) LDPC component codes, BP=10(9) and various scaling values, SS.

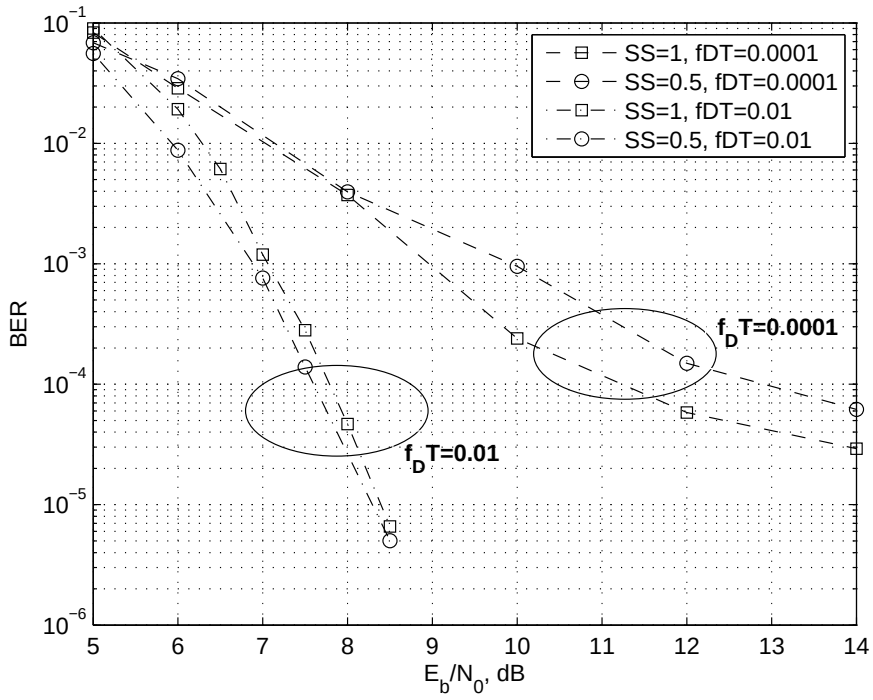
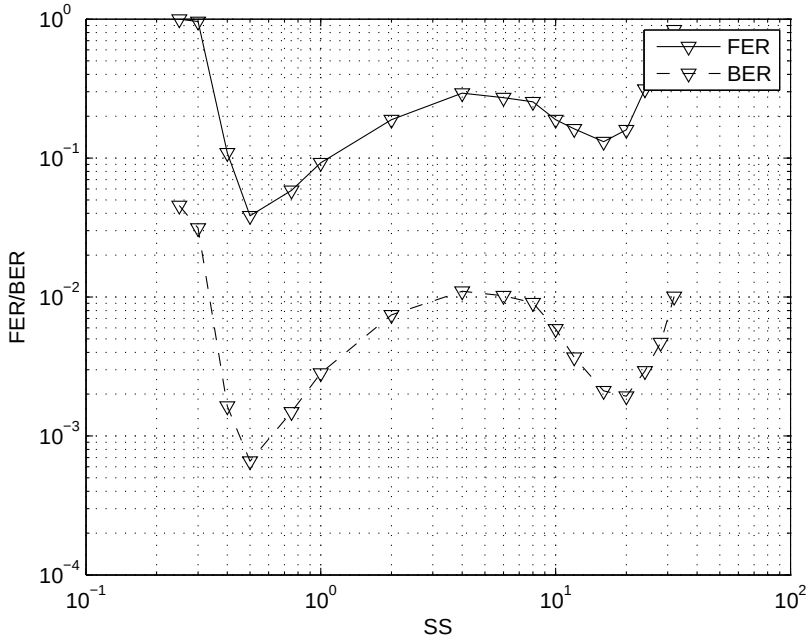


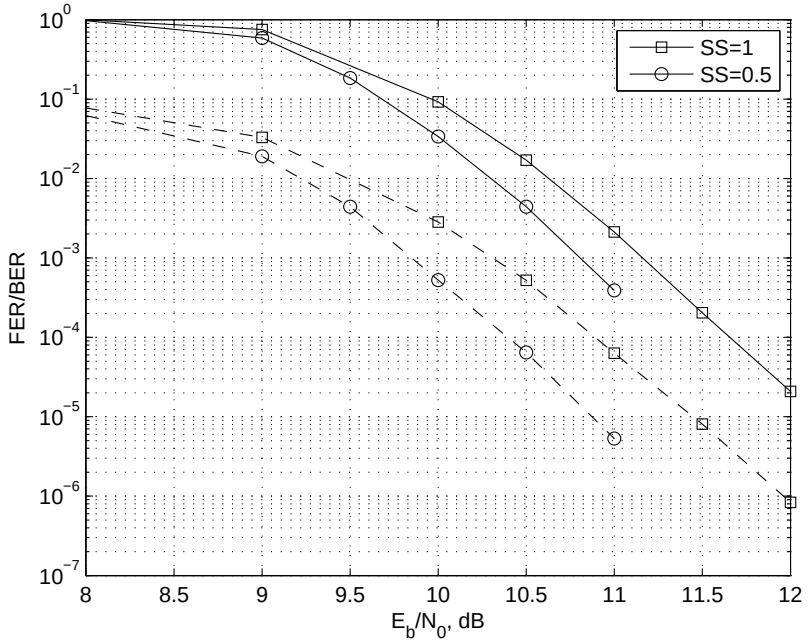
Figure 7.14: BER performance for the QPSK-R2-PIC system in the NoQS-fDT channel, for f_DT=0.01 and 0.0001, using SS=0.5 and 1, and (343,186) LDPC component codes. A CSI estimation error with $\rho = 0.99$ is assumed. Throughput is 2 bits/symbol period.

Fig. 7.14 shows the BER performance of the same system in both fast and slow fading across a wide SNR range. This shows that although the results in Fig. 7.13 were measured at one particular SNR value, they appear robust and can be used over a wide range of SNR. At BER=10^{-4}, a 0.2dB gain can be achieved by reducing SS from 1 to 0.5 for a fast-fading channel. In slow fading, a 1.5dB loss is observed.

We now increase the constellation size from QPSK to 16-QAM, giving us a throughput of 4.3 bits/symbol period. We perform our SS optimization for the QS-Ind channel used throughout Chapter 6, where perfect CSI is assumed. Fig. 7.15a shows that setting $SS = 0.5$ produces marginally the best FER and BER performances. This is expected since the QS-Ind channel effectively is a fast-fading channel. By setting $SS = 0.5$, Fig. 7.15b shows that a further 0.6dB performance improvement can be achieved over the 16QAM-R2-PIC system using $SS = 1$ at BER=10^{-5}.



(a) Varying SS values at 10dB.



(b) Varying SNR values, for SS=0.5 and 1.

Figure 7.15: FER (solid lines) and BER (dashed lines) performances for the 16QAM-R2-PIC system using (343,186) LDPC component codes and BP=10(9). Perfect CSI is assumed. Throughput is 4.3 bits/symbol period.

7.4 Summary

In this chapter, we have extended our simulations results to more realistic scenarios. First we have shown that the system still performs well when channel estimation is imperfect. A small diversity and performance loss occurs but no error floors are observed down to $BER = 10^{-6}$. Then we removed the quasi-static assumption and showed that the QPSK-R2-PIC system can outperform the 16QAM-R1-JD system in both fast and slow time-varying fading channels. Finally, we showed that performance can be further improved by optimally scaling the bit metrics passed to the BP decoders.

Chapter 8

Conclusion and Future Work

8.1 Accomplishments

In this thesis, we have developed a simple but powerful space-time coding system which simultaneously achieves

- high throughput,
- good performance,
- low processing delay,
- detection/decoding schemes with feasible complexity.

Currently, most existing systems only meet two or three of these four criteria. At the transmitter, we proposed the use of BMCM together with high rate QOST-BCs to obtain high system throughput. The parallel architecture of BMCM reduces processing delay and also allows greater design flexibility compared to the widely used serial architecture of BICM. At the receiver, we propose the use of PIC, which has low complexity, to decode the high throughput BMCM-QOSTBC scheme. Iterative PIC is a powerful tool for minimizing the effects of co-channel interference, when used together with powerful LDPC codes. Although suboptimal, PIC is shown to outperform the optimal JD detection scheme. This is due to its ability to exploit the code diversity of the LDPC codes repeatedly, which is not possible with JD.

We also developed a simple and novel LDPC code construction which generates LDPC codes with girth at least 6. This allows the low complexity BP algorithm

to be used for decoding. These codes were shown to perform as well as other existing LDPC codes in the literature. This code construction generates LDPC codes with different block lengths. For a given block length, we are able to generate LDPC codes with different code rates. This flexibility in code rate has two advantages:

- It allows the tradeoff between code rate and minimum distance. This enables us to design systems for a given error rate performance or system throughput.
- It also allows different code rates to be used for high order modulation, in conjunction with BMCM. This enables us to match different code rates to sublabels with different error protection, providing maximum throughput for any required error-rate.

Simulation results show that we can achieve a throughput of 5.6 bits/symbol period using $n_T = n_R = 4$ and 16-QAM and still maintain very good FER and BER performance, when perfect CSI is available and quasi-static fading is assumed. Despite the introduction of more interferers in symbol detection, using a high rate QOSTBC enables the use of a smaller modulation scheme, for a fixed throughput, which provides better overall noise protection. When the quasi-static assumption is removed, the proposed system maintains good performance in fast to slow fading ($f_D T = 0.01 - 0.001$). The results also show that the proposed system is robust to channel estimation errors, with the QPSK-R2-PIC scheme losing only 2.5dB in the presence of a 23% channel estimation error, with no diversity loss.

8.2 Future Work

Many extensions to the work in this thesis are possible. Some of them are listed in this section.

8.2.1 Row and column removals from $\boldsymbol{H}$

In Chapter 4, we developed a simple LDPC code construction. In Section 4.4, we showed how different code rates can be obtained for the same block length. This flexibility allows us to trade-off system throughput against performance. Code rate is increased by removing rows from $\boldsymbol{H}$, starting from the bottom row and moving

upwards. However, this may not be the optimum way of removing rows from $\boldsymbol{H}$. For the same number of rows to be removed, it is also possible to remove these rows from different parts of $\boldsymbol{H}$, rather than removing them all from the bottom part. Many different permutations exist and a further study to find the best permutation may result in better throughput versus performance tradeoff.

In contrast to row removals, column removals may also be performed on $\boldsymbol{H}$ to obtain a wide range of block lengths. This allows even greater flexibility in system design. It also increases the chances of making a fair comparison to other existing codes due to the ability to match the (N,K) code paramaters as close as possible.

8.2.2 Expanding $\boldsymbol{H}$ to increase girth

LDPC codes are widely used due to their capacity-approaching performance using the low complexity BP algorithm [46, 45]. However it is well known that BP decoding performs significantly worse in the presence of short cycles in the parity check matrix, especially those of length 4 [47, 48, 49]. Hence, BP decoding cannot be applied to many useful algebraic codes such as Golay codes and Hamming codes.

In [60], a necessary condition for the existence of a 4-cycle free Tanner graph was developed. In Section 4.6.1, we showed that the algebraic (7,3,4) dual Hamming code satisfies this condition. If we perform BP decoding on the systematic $\boldsymbol{H}$ of this code, the 4-cycles will degrade performance. However we showed how BP decoding can be performed on an expanded $\boldsymbol{H}$ with no 4-cycles. This expanded $\boldsymbol{H}$ is obtained using the V-construction developed in Section 4.2.

Further research on how to generalize the expansion of the $\boldsymbol{H}$ matrices to remove 4-cycles without altering the code properties is needed. This would allow powerful algebraic codes to be decoded using the low complexity BP algorithm.

8.2.3 Optimizing SS in BP bit metrics

In Section 7.3, we showed how performance can be further improved by optimizing the metric scaling factor SS when calculating bit metrics to the BP decoders. As discussed in [71], when the soft bit metrics are too large, polarization occurs and the soft information is effectively lost. This degrades performance. On the other hand,

when the soft bit metrics are too small, this effectively adds noise to the channel because we are effectively telling the BP decoders that there is an equal chance that each received bit is a "0" or a "1". This also degrades performance.

Our ad-hoc optimizations show that setting $SS = 0.5$ is optimal in QS-Ind channels and also in fast-fading NoQS-fDT channels. Setting $SS = 1$ is optimal in slow-fading NoQS-fDT channels. It would be interesting to investigate the optimal set of statistics to be passed to the BP decoders. This may depend upon the constellation used, LDPC code used, or even SNR. Once these statistics are known, we can systematically optimize the performance of any coded system employing LDPC codes.

8.2.4 Channel Estimation

The effect of CSI estimation errors was shown in Section 7.1. Channel estimation may be performed using data bits as well as pilot bits. Therefore, any scheme which iteratively provides improved estimates of the data, such as the PIC-BP decoding scheme, can use those estimates to provide improved channel estimates.

Appendix A

SNR Definition and Noise Variance Derivation

It is necessary to provide a fair comparison between our work and what is currently in the literature. In communications, this comparison is usually carried out using error-rate performance curves. These error-rate performance curves depend on the signal-to-noise ratio (SNR) used, which in turn depends on the transmit signal energy and the noise energy. To make a fair comparison, different systems should be normalized to the energy spent on each information bit transmitted. This sounds trivial, but communications is a very active research area which looks at many different channel conditions - additive white Gaussian noise (AWGN) channel, single-input-single-output (SISO) and multiple-input-multiple-output (MIMO) fading channels to name a few. The definition of SNR for a SISO system becomes ambiguous when used in a MIMO system. The SNR can be defined at each receive antenna or at the output of the diversity combiner. We need to know how these different SNR definitions relate to each other.

In Section A.1 we show how different SNR definitions are related to each other. In Section A.2 a unified noise variance which can be used in any 2^M-ary SISO, SIMO, MIMO system, with or without FEC coding, is derived. In Section A.3, we present some performance curves and we show which SNR definition provides a fairer comparison.

A.1 Different SNR Definitions

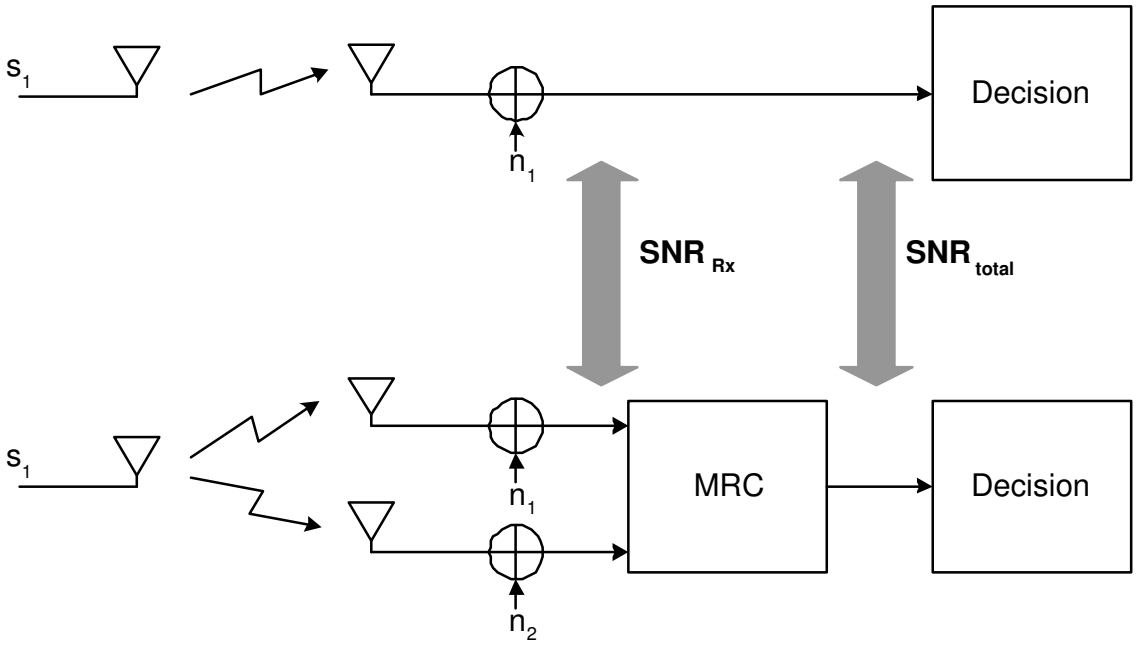


Figure A.1: Illustration of how the SNR per receive antenna, SNR_{Rx}, and the post-processing SNR, SNR_{total}, are defined in SISO (top) and SIMO (bottom) systems.

Fig. A.1 shows a SISO system at the top and a dual-diversity SIMO system at the bottom. We adopt maximum ratio combining (MRC) as it is widely known as the optimal receive diversity combining technique [15, 14]. The locations where the SNR per receive antenna, SNR_{Rx}, and the post-processing SNR, SNR_{total}, are defined are indicated in both systems. It is clear that in the SISO system, $SNR_{total} = SNR_{Rx}$. Therefore, a general SNR term is sufficient for a SISO system.

For a SIMO system, received signals at the two receive antennas are given by

$$r_1 = \alpha_1 s_1 + n_1 \qquad \text{and} \qquad r_2 = \alpha_2 s_1 + n_2 \tag{A.1}$$

where α_q is the fading coefficient between the transmit antenna and the q^{th} receive antenna and n_q is the corresponding AWGN receiver noise. The MRC block then produces the following estimate of s_1

$$\hat{s_1} = \alpha_1^* r_1 + \alpha_2^* r_2 \tag{A.2}$$

where $(.)^*$ denotes the conjugate transpose. Substituting equation (A.1) into equa-

tion (A.2) gives

$$\hat{s_1} = (|\alpha_1|^2 + |\alpha_2|^2)s_1 + \alpha_1^* n_1 + \alpha_2^* n_2. \tag{A.3}$$

The desired signal terms $|\alpha_1|^2 s_1$ and $\alpha_2^2 s_1$ add coherently whereas the noise terms $\alpha_1^* n_1$ and $|\alpha_2|^* n_2$ add incoherently. Therefore, the total energy of the signal component $\hat{s_1}$ is

$$(|\alpha_1|^2|s_1| + |\alpha_2|^2|s_1|)^2 = (|\alpha_1|^2 + |\alpha_2|^2)^2 E_s \tag{A.4}$$

since $|s_1| = \sqrt{E_s}$ and the total noise energy is

$$|\alpha_1|^2\sigma^2 + |\alpha_2|^2\sigma^2 = (|\alpha_1|^2 + |\alpha_2|^2)\sigma^2. \tag{A.5}$$

For fair comparison, we normalize $E_s = 1$. This gives an instantaneous SNR_{total} of $\frac{|\alpha_1|^2+|\alpha_2|^2}{\sigma^2}$. Since α_1 and α_2 are complex random variables representing a fading channel, we are more interested in the average SNR_{total}. We denote the expectation of α_k^2 by $E[|\alpha_k|^2] = \alpha^2$ for $k = 1, 2, ..., n_T$. Therefore, the average SNR_{total} for the dual-diversity SIMO system in Fig. A.1 is $\frac{2\alpha^2}{\sigma^2}$. It can easily be shown that the average SNR_{total} for a $(1, n_R)$ SIMO system employing MRC is $\frac{n_R\alpha^2}{\sigma^2}$.

At each receive antenna, the average signal energy is $(\alpha^2|s_1|)^2$ and the average noise energy is $\alpha^2\sigma^2$. Hence, the average SNR_{Rx} is $\frac{(\alpha^2)^2}{\alpha^2\sigma^2} = \frac{\alpha^2}{\sigma^2}$ for any SIMO system. Therefore, the relationship between the average SNR_{total} and the average SNR_{Rx} is given by

$$SNR_{total} = n_R SNR_{Rx}. \tag{A.6}$$

The SNR at the combiner output, SNR_{total}, is also known as the post-processing SNR. This is the SNR definition used throughout this thesis. In Section A.3, we show it is better to use SNR_{total}, as opposed to SNR_{Rx}.

A.2 Noise Variance Derivation

Here, we provide a step-by-step transformation of the well known definition of the noise variance for an uncoded binary SISO system into that for a generalized coded 2^M-ary MIMO system. Note that with each step, the definition of terms is gradually refined.

A.2.1 Uncoded Binary SISO

The one-dimensional noise variance for an uncoded binary SISO system is

$$\sigma^2 = \frac{N_0}{2} = \frac{E_b}{2 \cdot SNR} \tag{A.7}$$

where

$$SNR = \frac{E_b}{N_0} \tag{A.8}$$

is the ratio of signal energy to noise energy at the receive antenna. Each bit is mapped to one symbol for transmission. Therefore, the symbol energy E_s equals the bit energy E_b.

A.2.2 Coded Binary SISO

When (n, k) FEC codes are used, we need to make a distinction between information bits and codeword bits. A stream of k information bits are mapped onto n codeword bits. Therefore, each codeword bit has energy

$$E_c = \frac{kE_b}{n}. \tag{A.9}$$

Rearranging and substituting equation (A.9) into equation (A.7), the one-dimensional noise variance for a coded binary SISO system becomes

$$\sigma^2 = \frac{N_0}{2} = \frac{E_c}{2R_{fec} \cdot SNR} \tag{A.10}$$

where $R_{fec} = k/n$ is the FEC code rate and E_c is the codeword bit energy. The n codewords bits are then mapped onto n symbols for transmission. Therefore, the symbol energy E_s is the same as E_c.

A.2.3 Coded 2^M-ary SISO

In a coded binary system, each codeword bit selects a symbol to be transmitted. For a coded 2^M-ary system, M codeword bits selects one symbol to be transmitted. Therefore, $E_s = ME_c$. Substituting this into equation (A.10), the one-dimensional noise variance for a coded 2^M-ary SISO system becomes

$$\sigma^2 = \frac{N_0}{2} = \frac{E_s}{2MR_{fec} \cdot SNR} \tag{A.11}$$

where E_s is now the average[1] transmitted symbol energy.

[1] In larger constellations, not every symbol contains equal energy. Therefore, the average energy across all symbols in any constellation is used.

A.2.4 Coded 2^M-ary SIMO

For SISO systems, the SNR is usually defined at the point where the noise is added. This happens at the receive antenna. Therefore, the usual SNR term is actually the SNR_{Rx} term in Fig. A.1. We can rewrite equation (A.11) as

$$\sigma^2 = \frac{N_0}{2} = \frac{E_s}{2MR_{fec} \cdot SNR_{Rx}}. \tag{A.12}$$

The SNR_{Rx} term is the same as the SNR immediately before making decisions, SNR_{total} (c.f. Fig. A.1). Therefore, the usual SNR term is sufficient to determine the noise variance for SISO systems.

For SIMO systems, $SNR_{total} = n_R SNR_{Rx}$, as shown in equation (A.6). Therefore, a distinction needs to be made between SNR_{Rx} and SNR_{total}. Substituting equation (A.6) into equation (A.12), the one-dimensional noise variance for a coded 2^M-ary SIMO system becomes

$$\sigma^2 = \frac{N_0}{2} = \frac{n_R E_s}{2MR_{fec} \cdot SNR_{total}} \tag{A.13}$$

A.2.5 Coded 2^M-ary MIMO

For single-transmit-antenna systems, the symbol generated by the modulator block is transmitted directly[2] over the channel. For multiple-transmit-antenna systems in this thesis, STBCs are used. This is illustrated in Fig. A.2 using Alamouti's $n_T = 2$ OSTBC [8], Tarokh's 1/2-rate $n_T = 4$ OSTBC [24], a 3/4-rate $n_T = 4$ OSTBC [18], Jafarkhani's Rate 1 $n_T = 4$ QOSTBC [17] and the Rate 2 $n_T = 4$ QOSTBC [16]. In Fig. A.2a, the OSTBC takes two new symbols and generates four output symbols to be transmitted over the channel using two antennas and two time slots. In Fig. A.2b, the OSTBC takes 4 new symbols and generates 32 output symbols to be transmitted over the channel using four antenna and eight time slots. In Fig. A.2c, the OSTBC takes 3 new symbols and generates 12 output symbols to be transmitted, using four antennas and four time slots. However, in any time slot, only three antennas are transmitting - the remaining antenna is turned off. In Fig. A.2d, the Rate 1 QOSTBC takes 4 new symbols and generates 16 output symbols to be transmitted, using four antennas and four time slots. In Fig. A.2e, the Rate 2

[2]The STBC for single transmit antenna systems can be considered a simple $[\boldsymbol{s_i}]$ matrix. It takes one new symbol $\boldsymbol{s_i}$ and transmits $\boldsymbol{s_i}$ over one time slot.

QOSTBC takes 4 new symbols and generates 8 output symbols to be transmitted, using four antennas and two time slots.

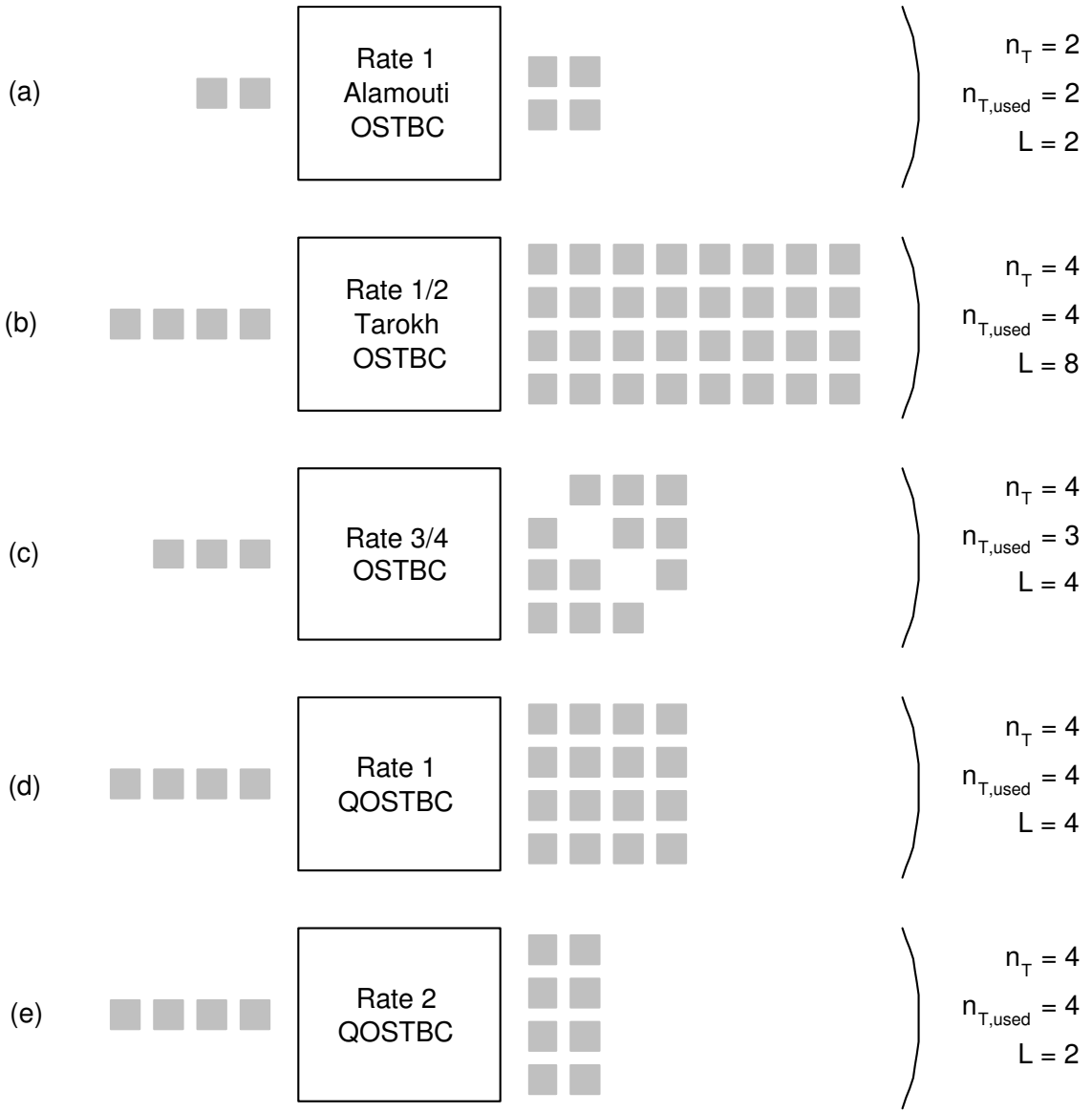


Figure A.2: Different STBC schemes for $n_T = 2$ and 4 transmit antennas.

Let w denote the number of new symbols required by the STBC. Let L be the number of time slots required to transmit all the symbols generated by the STBC. For multiple transmit antenna systems, we need to distinguish the symbols generated by the modulator (input to STBC) from the symbols generated by the STBC for transmission. We can then generalize the examples in Fig. A.2. Any STBC takes w new symbols and transmits $n_{T,used}L$ symbols in total over the channel, where $n_{T,used}$ is the number of antennas used for transmission within each time slot and

$n_{T,used} \leq n_T$. Let E_s be the symbol energy generated by the modulator, as in the single transmit antenna cases, and $E_{s,Tx}$ be the actual transmitted symbol energy from each antenna in each time slot. The total energy on both sides of the STBC block must be the same. Therefore, $wE_s = n_{T,used}LE_{s,Tx}$ and rearranging this, we have

$$E_s = \frac{n_{T,used}LE_{s,Tx}}{w}. \tag{A.14}$$

Substituting equation (A.14) into equation (A.13), the one-dimensional noise variance for a coded 2^M-ary MIMO system becomes

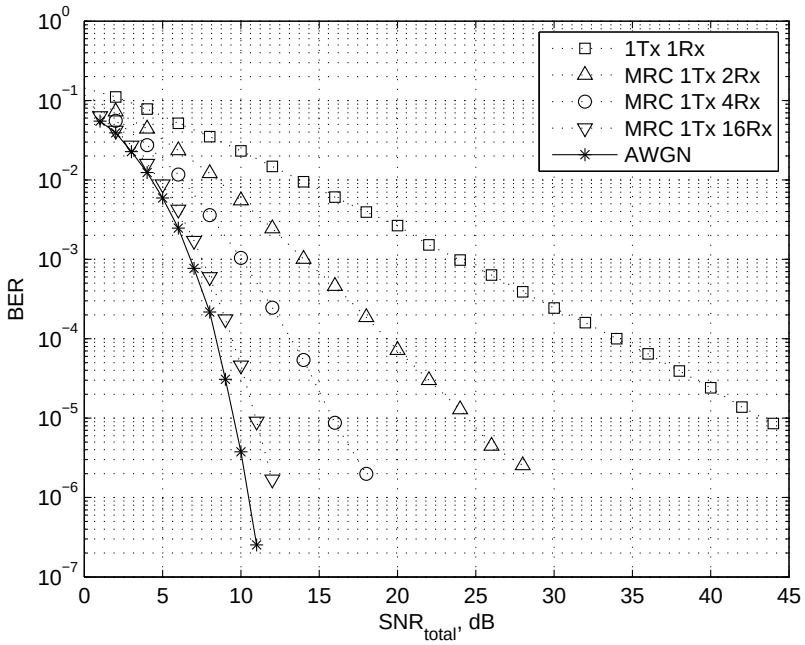
$$\sigma^2 = \frac{N_0}{2} = \frac{n_{T,used}n_R E_{s,Tx}}{2MR_{stbc}R_{fec}SNR_{total}}, \tag{A.15}$$

where $R_{stbc} = w/L$ is the rate of the STBC as defined in equation (2.2).

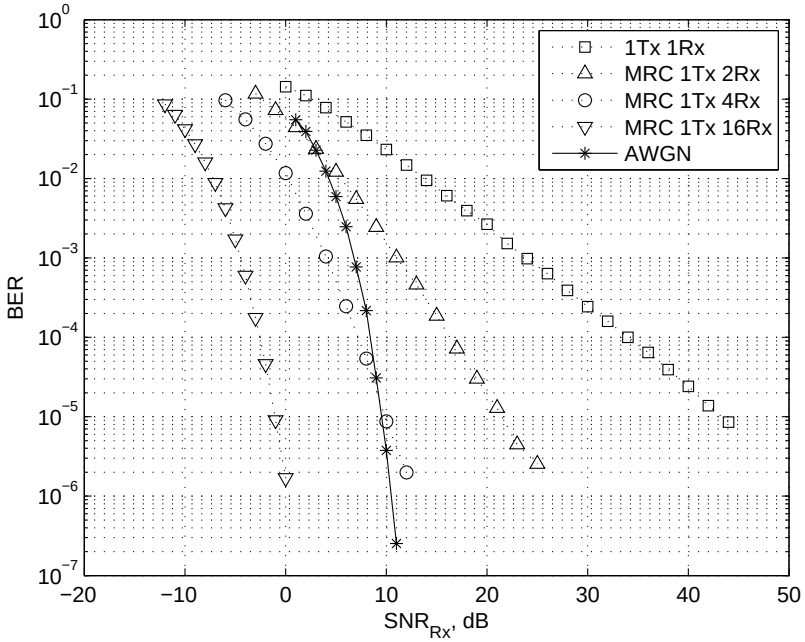
A.3 Using SNR_{total} versus using SNR_{Rx}

Using Monte Carlo simulations, we show the standard performance curves in Rayleigh fading channel. MRC schemes are employed for multiple receive antennas. We compare the performance curves obtained using SNR_{total} versus SNR_{Rx}, and the noise variance as derived in Section A.2.

Fig. A.3a shows the performance in Rayleigh fading using QPSK, the post-processing SNR, SNR_{total}, and the noise variance from equation (A.13). Fig. A.3b shows the same set of curves but using the SNR per receive antenna, SNR_{Rx}. Comparing Fig. A.3b to Fig. A.3a, it is clear that by using SNR_{Rx}, the performance curves appear significantly better. The apparent gain from using SNR_{Rx} over SNR_{total} is about $10log_{10}n_R$ dB. In Fig. A.3b, the Rayleigh faded SIMO system using 16 receive antennas appears to achieve better performance than in the AWGN channel. This does not reflect the actual effect of employing multiple receive antennas, which is to mitigate the effects of fading, thereby transforming the Rayleigh fading channel into essentially an AWGN channel [42] as n_R increases. However, the transformation of the Rayleigh fading channel into an AWGN through the addition of receive antennas is clearly depicted in Fig. A.3a which uses SNR_{total}.



(a) Horizontal axis uses SNR_{total}.



(b) Horizontal axis uses SNR_{Rx}.

Figure A.3: Standard performance curves in Rayleigh fading using QPSK for $n_T = 1$ and $n_R = 1, 2, 4, 16$. The AWGN performance is also shown.

A.4 Summary

In this appendix, we have carefully derived and clearly defined the noise variance used in all our simulations. We also showed that although SNR per receive antenna may make simulations results appear better, post-processing SNR provides a more accurate picture of system performance.

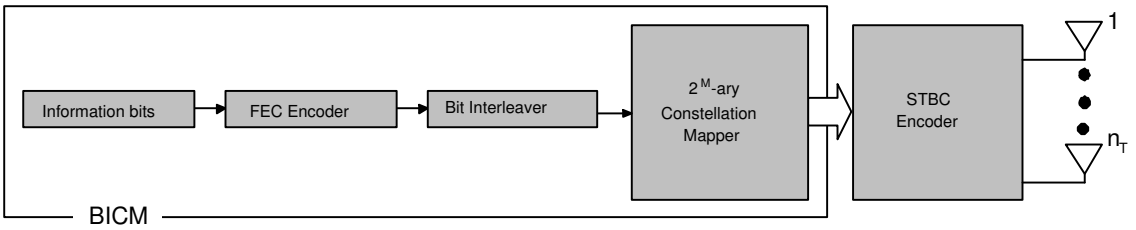
Appendix B

Coded Modulation Structures

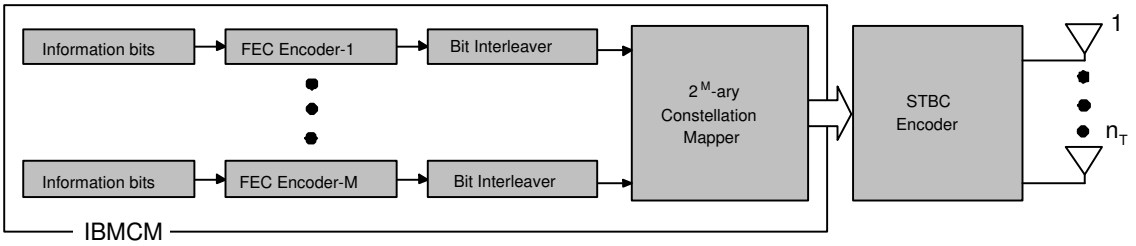
Coded modulation schemes are bandwidth efficient schemes that exploit time diversity on single transmit and receive antenna multipath fading systems [79]. In [35], Zehavi recognized that code diversity, which increases the reliability of coded modulation, can be improved by using bit interleaving. This is known as bit-interleaved coded modulation (BICM) [35, 34]. For multiple antenna systems, BICM structures can be serially concatenated with spatial multiplexing [36] or space-time codes [37]. We focus here on the use of space-time block codes (STBCs), as discussed in Chapter 2.

Fig. B.1a shows the serial concatenation of BICM with a STBC [37]. The information bit stream is first encoded using a forward error correcting (FEC) code. The coded bit stream is then bit interleaved and every group of M interleaved bits are used to select one of 2^M constellation symbols. Each block of L symbols is then transmitted over the channel using space-time codes, n_T transmit antennas and L time slots.

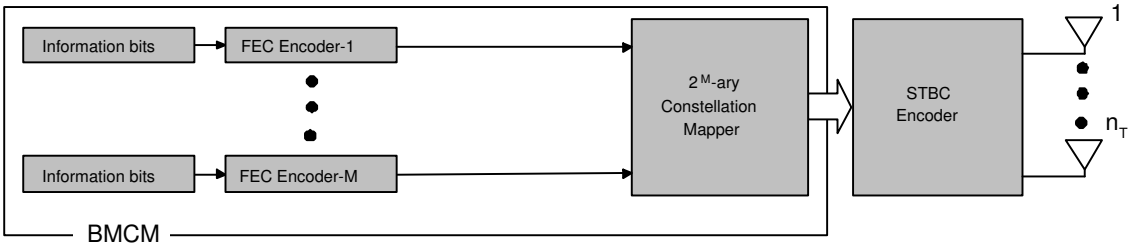
In Chapter 3, we introduced a coded modulation structure called interleaved bit-mapped coded modulation (IBMCM). For a constellation of size 2^M, Fig. B.1b shows that this structure is essentially M bit-interleaved coded modulation (BICM) [34, 35] structures in parallel. The information bit stream is first multiplexed into M substreams and each is encoded using a FEC code. A bit interleaver is then applied to each encoded substream. Instead of using groups of M encoded bits, IBMCM uses only one encoded bit from each of the M substreams to select one constellation symbol. Although this requires more hardware compared to BICM, it



(a) BICM-STBC



(b) IBMCM-STBC



(c) BMCM-STBC

Figure B.1: Coded modulation structures for multiple antenna systems.

reduces processing delays, which is important in voice transmission.

In Chapter 5, the bit interleavers are removed from IBMCM to produce a simpler bit-mapped coded modulation (BMCM) structure. This is illustrated in Fig. B.1c. The removal of the interleavers reduces processing delays even further.

Fig. B.2 compares the bit-error rate (BER) performance for the BICM, IBMCM and BMCM structures. Each structure is serially concatenated with the Alamouti OSTBC [8] and uses 16-QAM. The BICM structure uses the (1331,730) LDPC component codes from Chapter 4. The IBMCM and BMCM structures use (343,202) LDPC component codes on sub-labels[1] AB and (343,172) LDPC component codes on sub-labels CD. This provides an approximate system throughput of

[1]A discussion on unequal error correction for 16-QAM is provided in Section 6.6.

2.1 bit/symbol period for fair comparison. Independent quasi-static fading is assumed, where the fading coefficients remain constant for a block of $L = 2$ time slots and vary independently from block to block[2]. Perfect CSI is assumed at the receiver.

Fig. B.2 shows that the IBMCM structure outperforms BICM at low SNR, but BICM outperforms IBMCM at high SNR. At BER=10^{-5}, BICM outperforms IBMCM by about 0.9dB. For the same frame length (containing approximately the same number of information bits), the BICM structure is able to use a component code M times longer than the component codes in the IBMCM structure, for 2^M-ary constellations. This provides higher minimum distance, which explains its superior performance at high SNR. However, the processing delay required for IBMCM is only $\frac{1}{M}$ that of BICM. In addition, IBMCM allows parallel processing, which suits implementation using the increasingly popular field-programmable gate arrays (FPGAs) [80].

The underlying system uses the Alamouti OSTBC with $n_T = n_R = 2$, giving a diversity order of 4. Note that BICM only outperforms IBMCM when the BER is below 10^{-2}. When $n_T = n_R = 4$, the maximum diversity order increases to 16. We expect the increased diversity to benefit IBMCM more than BICM, as BICM achieves greater diversity with its longer component code. This is because the translation of diversity gain into SNR gain decreases with diversity order, as shown in Section 2.3. The (1331,730) LDPC component code used in the BICM structure is also the best performing length 1331 code, as shown in Fig. 4.7. The (343,202) and (343,172) LDPC component codes used in the IBMCM structure are not the best performing length 343 codes. They were selected to cater for the unequal error protection in 16-QAM and also to achieve the same overall system throughput as the BICM structure.

Fig. B.2 also shows that the BMCM and IBMCM structures perform equally well in both FER and BER. Therefore, interleaving is not required for the quasi-static fading channels considered in most literature. Although IBMCM has shorter processing delay compared to BICM, we can reduce this delay even further by employing BMCM. Therefore, the BMCM structure provides greater flexibility, as well

[2]For the length-343 codes used in the IBMCM or BMCM structures, each codeword experiences 172 independent fading coefficients. This represents a fast-fading scenario, as verified by the results in Chapter 7.

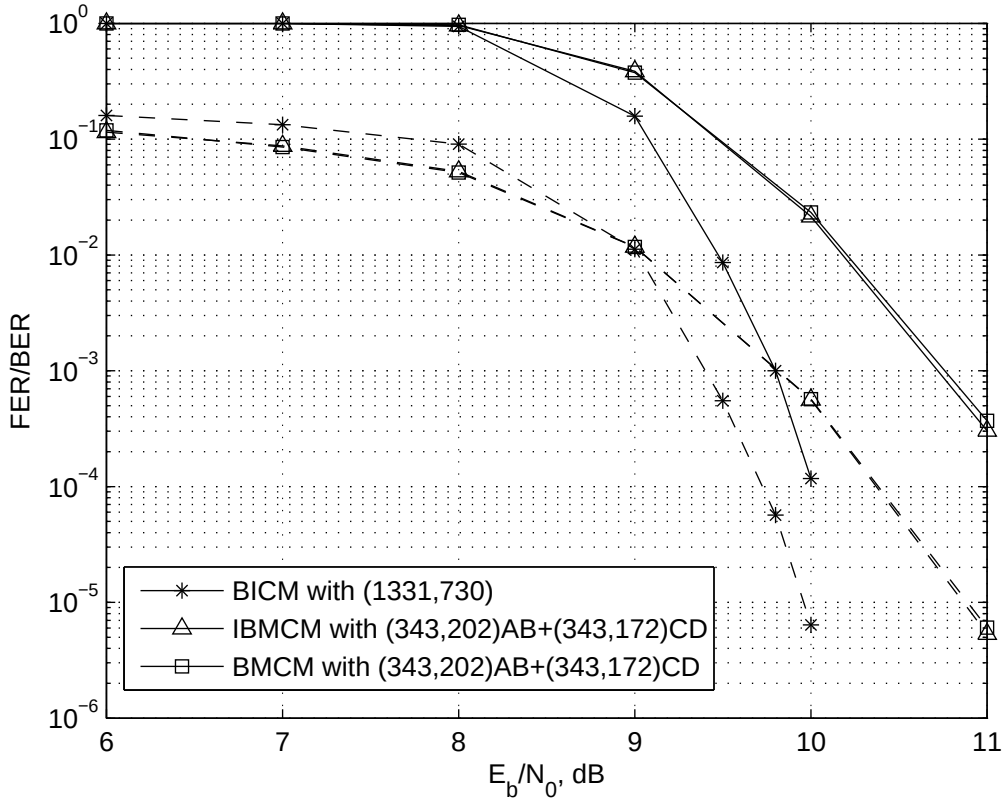


Figure B.2: FER (solid lines) and BER (dashed lines) performance for the BICM, IBMCM and BMCM structures. LDPC component codes are used with $n_T = n_R = 2$. Each structure is serially concatenated with the Alamouti OSTBC using 16-QAM, giving a system throughput of 2.1 bit/symbol period.

as reduced processing delay, compared to the BICM structure. This flexibility allows more freedom in system design tradeoffs for any given throughput or performance.

Appendix C

Proof for LDPC Girth

In Section 4.3, we introduced two new construction algorithms for low-density parity check (LDPC) codes. Here, we show that the Density-$\frac{p}{p^2}$ and Density-$\frac{p}{p^3}$ algorithms generate LDPC matrices of girth 6. The resulting parity check matrices from the Density-$\frac{p}{p^2}$ algorithm are similar to array codes [81], a fully structured class of LDPC codes. However, the Density-$\frac{p}{p^3}$ algorithm generates a new class of fully structured LDPC codes.

C.1 Array Codes

Array codes [81] are generally used in data storage due to their capability of detecting and correcting error bursts. This also indicates their use for quasi-static fading channels. Recently, they were generalized into LDPC codes [82] and later into quasi-cyclic LDPC (QC-LDPC) codes [83]. These codes are similar to the ones generated using the Density-$\frac{p}{p^2}$ algorithm. Array-based LDPC codes can be shortened by deleting columns of the parity-check matrices to increase their girth [84]. This is analogous to the trellis based removal of short cycles by column deletion in [49]. Instead of column removal, the Density-$\frac{p}{p^3}$ algorithm essentially adds columns and rows to the parity-check matrix of the Density-$\frac{p}{p^2}$ algorithm to increase girth.

An array or QC-LDPC code with a regular row weight p and column weight

q has the form [82, 84, 83]

$$\boldsymbol{H} = \begin{bmatrix} I & I & ... & I \\ I & \boldsymbol{S}^1 & ... & \boldsymbol{S}^{q-1} \\ : & : & & : \\ I & \boldsymbol{S}^{r-1} & ... & \boldsymbol{S}^{(r-1)(q-1)} \end{bmatrix} \tag{C.1}$$

where the circulant permutation matrix $\boldsymbol{S}$ is obtained by one cyclic shift of the q-by-q identity matrix. This can also be represented in a more general form [84]

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{S}^{a_0 \cdot 0} & \boldsymbol{S}^{a_0 \cdot 1} & ... & \boldsymbol{S}^{a_0 \cdot (q-1)} \\ \boldsymbol{S}^{a_1 \cdot 0} & \boldsymbol{S}^{a_1 \cdot 1} & ... & \boldsymbol{S}^{a_1 \cdot (q-1)} \\ : & : & & : \\ \boldsymbol{S}^{a_{p-1} \cdot 0} & \boldsymbol{S}^{a_{p-1} \cdot 1} & ... & \boldsymbol{S}^{a_{p-1} \cdot (q-1)} \end{bmatrix} \tag{C.2}$$

where $a_0, a_1, ..., a_{p-1}$ is some sequence of p distinct integers from[1] $[0, q-1]$. These are the p block-row labels used to define blocks of q rows in (C.2). Similarly, q distinct block-column labels are used to define blocks of q columns in (C.2). A closed path of length $2k$ in any parity-check matrix of the form in (C.2) is a sequence of block-row and block-column index pairs

$$(i_1, j_1), (i_1, j_2), (i_2, j_2), (i_2, j_3), ..., (i_k, j_k), (i_k, j_1) \tag{C.3}$$

with $i_l \neq i_{l+1}$, $j_l \neq j_{l+1}$, for l=1,2,...,$k-1$, and $i_k \neq i_1$, $j_k \neq j_1$.

C.2 Girth 6 Proof for the Density-$\frac{p}{p^2}$ Algorithm

The proof for a general girth of $2k$ for the array code can be found in [82, 84]. Here, we present the special case of girth 6 for the Density-$\frac{p}{p^2}$ algorithm. This algorithm generates an array code with $a_i = i$, for $0 < i < (p-1)$ and $p = q$.

Theorem 1: A cycle of length 4 exists in the Tanner graph of an array code with parity-check matrix $\boldsymbol{H}$ and labels $a_0, a_1, ..., a_{p-1}$ if and only if there exists a closed path

$$(i_1, j_1), (i_1, j_2), (i_2, j_2), (i_2, j_1) \tag{C.4}$$

in H such that

$$\boldsymbol{S}^{a_{i_1} \cdot j_1} (\boldsymbol{S}^{a_{i_1} \cdot j_2})^{-1} \boldsymbol{S}^{a_{i_2} \cdot j_2} (\boldsymbol{S}^{a_{i_2} \cdot j_1})^{-1} = I. \tag{C.5}$$

[1]The notation $[a, b]$ is used to denote the set $\{x \in Z : a \leq x \leq b\}$.

Since $\boldsymbol{S}$ is a q-by-q circulant permutation matrix, $\boldsymbol{S} \neq \boldsymbol{I}$, and q is prime, we can have $\boldsymbol{S}^n = \boldsymbol{I}$ if and only if $n \equiv 0 \pmod{q}$. Therefore, the condition in Theorem 1 is equivalent to

$$a_{i_1}(j_1 - j_2) + a_{i_2}(j_2 - j_1) \equiv 0 \pmod{q} \tag{C.6}$$

which can also be written as

$$(a_{i_1} - a_{i_2})(j_1 - j_2) \equiv 0 \pmod{q}. \tag{C.7}$$

This is impossible since $i_1 \neq i_2$ and $j_1 \neq j_2$. Hence, $0 < |a_{i_1} - a_{i_2}| < q$ and $0 < |j_1 - j_2| < q$. Therefore, the $\boldsymbol{H_2}$ matrices[2] generated using the Density-$\frac{p}{p^2}$ algorithm have girth at least 6.

C.3 Girth 6 Proof for the Density-$\frac{p}{p^3}$ Algorithm

The $\boldsymbol{H_3}$ matrices generated by the Density-$\frac{p}{p^3}$ algorithm may be viewed as a Kronecker product of $\boldsymbol{H_2}$ with $\boldsymbol{S}^{b_{i,j}}$ where $\boldsymbol{S}$ is a circulant permutation matrix of size q-by-q, $b_{i,j} = [0, q]$, and (i, j) is the (row,column) label for $\boldsymbol{H_2}$. The idea of designing LDPC codes using Kronecker products is first observed in [85], where the core matrix $\boldsymbol{H}(0)$ was found by computer search based on a girth criterion. At each recursive step m, a "1" in the parity check matrix $\boldsymbol{H}(m-1)$ is replaced by a random permutation matrix. In [86], the core matrix is a structured QC-LDPC, but the Kronecker product is again performed using pseudorandom permutation matrices. In our LDPC codes, a "1" is replaced by a deterministic permutation matrix.

Fig. C.1 illustrates how the Kronecker product affects the girth of a matrix. We consider the 3-by-3 matrix $\boldsymbol{H}$ in Fig. C.1a which is defined by a single cycle of length 6. Fig. C.1b shows that the Kronecker product of $\boldsymbol{H}$ with a 2-by-2 identity matrix produces 2 independent cycles of length 6. In general, the Kronecker product of $\boldsymbol{H}$ with $\boldsymbol{S}^{b_{i,j}}$, where $b_{i,j}$ is constant for all i and j, produces q independent cycles of girth($\boldsymbol{H}$). If $b_{i,j}$ is not constant for all i and j values, as illustrated in Fig. C.1c, this could break up the cycles in Fig. C.1b, resulting in a new girth greater than

[2]In this appendix, we are providing the relationship between the Density-$\frac{p}{p^3}$ and Density-$\frac{p}{p^3}$ algorithms. Therefore, we use the notations $\boldsymbol{H_2}$ and $\boldsymbol{H_3}$ to denote the parity check matrix generated by each algorithm, respectively.

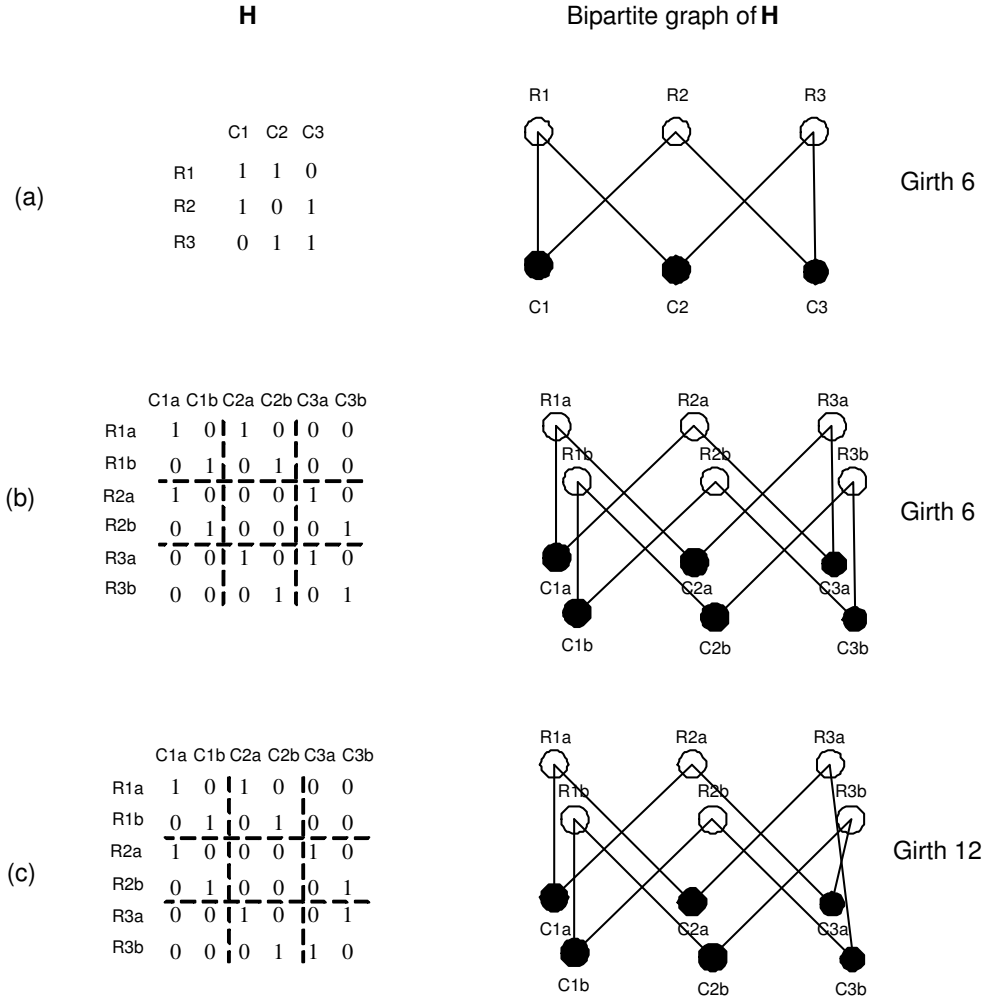


Figure C.1: How the Kronecker product affects girth: (a) original $\boldsymbol{H}$ with girth 6; (b) $\boldsymbol{H} \otimes \boldsymbol{S}^{w_{i,j}}$ where $w_{i,j} = 0$ for all i, j; and (c) $\boldsymbol{H} \otimes \boldsymbol{S}^{w_{i,j}}$ where $w_{i,j} = [0, q]$.

or equal to girth($\boldsymbol{H}$). As long as the Kronecker product is performed using circulant permutation matrices[3], the girth cannot be reduced. This is because any two adjoining q-by-q identity matrices, regardless of any row and column permutations, are represented by q disjointed open lines of length 2 in the bipartite graphs.

Since girth($\boldsymbol{H_2}$)$\geq$6, and the Kronecker product of $\boldsymbol{H_2}$ with circulant permutation matrices does not reduce girth, we can prove that the parity-check matrices obtained using the Density-$\frac{p}{p^3}$ algorithm also have girth at least 6.

[3]This is true for any row and/or column permutations of identity matrices. A circulant permutation matrix is a subset of such permutations.

Bibliography

[1] P. Chaudhury, "The 3GPP proposal for IMT-2000," *IEEE Commun. Mag.*, vol. 37, pp. 72–81, Dec 1999.

[2] PracTel Inc., "4G: Assessment markets and technology." http://www. researchandmarkets.com/reports/c38975, June 2006.

[3] I. E. Telatar, "Capacity of multi-antenna gaussian channels," *European Trans. Telecommun.*, vol. 10, Nov.-Dec. 1999.

[4] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, Mar 1998.

[5] G. J. Foschini, "Layered space-time architecture for wireless communication is a fading environment when using multiple antennas," *Bell Labs. Tech. J.*, vol. 1, autumn 1996.

[6] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. 44, pp. 744–764, Mar 1998.

[7] V. Tarokh, A. Naguib, N. Seshadri, and A. R. Calderbank, "Combined array processing and space-time coding," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1121–1128, May 1999.

[8] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1451–1458, Oct 1998.

[9] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, pp. 372–423,623–656, 1948.

[10] W. Firmanto, J. Yuan, K. L. Lo, and B. Vucetic, "Layered space-time coding: Performance analysis and design criteria," *in Proc. Globecom*, vol. 2, pp. 1083–1097, Nov. 2001.

[11] J. Du and Y. G. Li, "Parallel detection of space-time codes by predictive soft interference cancellation," *in Proc. ICC*, vol. 5, pp. 2746–2750, June 2004.

[12] S. Benedetto and E. Biglieri, *Principles of Digital Transmission with Wireless Applications*. Kluwer Academic/ Plenum Publishers, 1999.

[13] S. Haykin, *Communication Systems*. John Wiley & Sons, Inc., 3 ed., 1994.

[14] T. S. Rappaport, *Wireless communications - principles and practise*. Prentice Hall, 1996.

[15] J. G. Proakis, *Digital communications*. McGraw-Hill, 3rd ed., 1995.

[16] E. N. Onggosanusi, A. G. Dabak, and T. A. Schmidl, "High rate space-time block coded scheme: Performance and improvement in correlated fading channels," *in Proc. WCNC*, vol. 1, pp. 194–199, Mar. 2002.

[17] H. Jafarkhani, "A quasi-orthogonal space-time block code," *IEEE Trans. Commun.*, vol. 49, pp. 1–4, Jan. 2001.

[18] H. Lee, M. Siti, W. Zhu, and M. Fitz, "Super-orthogonal space-time block code using a unitary expansion," *in Proc. VTC-Fall*, vol. 4, pp. 2513–2517, Sept. 2004.

[19] T. H. Liew and L. Hanzo, "Space-time codes and concatenated channel codes for wireless communications," *Proc. IEEE*, vol. 90, pp. 187–219, Feb. 2002.

[20] A. Stefanov and T. M. Duman, "Turbo-coded modulation for systems with transmit and receive antenna diversity over block fading channels: System model, decoding approaches, and practical considerations," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 958–969, May 2001.

[21] S. H. Muller-Weinfurtner, "Coding for multiple antenna transmission in fast fading and in OFDM," *in Proc. ICC*, vol. 3, pp. 1691–1695, 2002.

[22] B. M. Hochwald and S. T. Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, pp. 389–399, Mar. 2003.

[23] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with soft detection based on hard sphere decoding," *IEEE Trans. Commun.*, vol. 4, pp. 587–590, Apr. 2006.

[24] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1456–1467, July 1999.

[25] M. Rupp and C. F. Mecklenbruker, "On extended alamouti schemes for space-time coding," *in Proc. WPMC*, vol. 1, pp. 115–119, Oct. 2002.

[26] S.-Y. Chung, G. D. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *IEEE Commun. Lett.*, vol. 5, pp. 58–60, Feb. 2001.

[27] M. Schwartz, W. R. Bennett, and S. Stein, *Communication Systems and Techniques.* McGraw Hill, 1966.

[28] B. Manatsavee, K. Ahmed, and A. Fernando, "Performance of PIC,SIC and decorrelating detectors for MUD technique in WCDMA system," *in Proc.* ICICS-PCM, vol. 2, pp. 892–896, Dec. 2003.

[29] N. Benvenuto and P. Bisaglia, "Parallel and successive interference cancellation for MC-CDMA and their near-far resistance," *in Proc. VTC-Fall*, vol. 2, pp. 1045–1049, Oct. 2003.

[30] P. Patel and J. Holtzman, "Performance comparison of a DS/CDMA system using a successive interference cancellation (IC) scheme and a parallel IC scheme under fading," *in Proc. ICC*, vol. 1, pp. 510–514, May 1994.

[31] H. K. Yeo, B. S. Sharif, O. R. Hinton, and A. E. Adams, "Analysis of a multi-element multi-user receiver for a shallow water acoustic network (SWAN) based on a recursive successive interference cancellation (RSIC)technique," *in Proc. OCEANS*, vol. 3, pp. 1537–1541, Sept. 1999.

[32] T. A. Tran, T. X. Lai, and A. B. Sesay, "Single-carrier concatenated space-time block coded transmissions over selective-fading channels," *in Proc. Canadian Conference on Electrical and Computer Engineering*, vol. 3, pp. 1577–1580, May 2004.

[33] S. B. Wicker, *Error control systems for digital communication and storage.* Prentice Hall, 1995.

[34] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1998.

[35] E. Zehavi, "8-PSK trellis codes for a rayleigh channel," *IEEE Trans. Commun.*, vol. 40, pp. 873–884, May 1992.

[36] A. M. Tonello, "Space-time bit-interleaved coded modulation with an iterative decoding strategy," *in Proc. VTC-Fall*, vol. 1, pp. 473–478, Sept. 2000.

[37] Z. Hong and B. L. Hughes, "Bit-interleaved space-time coded modulation with iterative decoding," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 1912–1917, Nov. 2004.

[38] J. S. K. Tee and D. P. Taylor, "Suboptimal SISO decoding of systematic binary algebraic block codes," *IEEE Trans. Commun.*, vol. 51, pp. 715–718, May 2003.

[39] R. van Nobelen and D. P. Taylor, "Analysis of the pairwise error probability of noninterleaved codes on the rayleigh-fading channel," *IEEE Trans. Commun.*, vol. 44, pp. 456–463, Apr. 1996.

[40] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technology*, vol. 19, pp. 751–772, Oct 1971.

[41] C. L. H. Lee, *Convolutional Coding: Fundamentals and Applications.* Artech House, 1997.

[42] G. Bauch and J. Hagenauer, "Analytical evaluation of space-time transmit diversity with FEC-coding," *in Proc. Globecom*, vol. 1, pp. 435–439, Nov 2001.

[43] R. G. Gallager, *Low Density Parity Check Codes.* Cambridge, Mass.: MIT Press, 1963.

[44] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error correcting coding and decoding: Turbo-codes (1)," *in Proc. ICC*, pp. 1064–1070, 1993.

[45] D. J. C. MacKay and R. M. Neal, *Good Codes Based on Very Sparse Matrices*, vol. 1025. Crytography and Coding. 5th IMA Conf., 29th August 1995.

[46] S. Lin and D. J. Costello Jr, *Error control coding*. Pearson Prentice Hall, 2 ed., 2004.

[47] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[48] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[49] L. Lan, Y. Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar, "A trellis-based method for removing cycles from bipartite graphs and construction of low density parity check codes," *IEEE Commun. Lett.*, vol. 8, pp. 443–445, July 2004.

[50] J. Campello, D. S. Modha, and S. Rajagopalan, "Designing LDPC codes using bit-filling," *in Proc. ICC*, vol. 1, pp. 55–59, June 2001.

[51] Y. Mao and A. H. Banihashemi, "A heuristic search for good LDPC codes at short block lengths," *in Proc. ICC*, vol. 1, pp. 41–44, June 2001.

[52] Y. Kou, S. Lin, and M. Fossorier, "Construction of low density parity check codes: A geometric approach," *in Proc. 2nd Int. Symp. Turbo Codes and Related Issues*, pp. 137–140, Sept. 2000.

[53] D. H. Johnson and G. C. Orsak, "Relation of signal set choice to the performance of optimal non-gaussian detectors," *IEEE Trans. Commun.*, vol. 41, pp. 1319–1328, sep 1993.

[54] J. M. F. Moura, J. Lu, and H. Zhang, "Structured low-density parity-check codes," *IEEE Signal Processing Magazine*, pp. 42–55, Jan. 2004.

[55] L. Wei, "Several properties of short LDPC codes," *IEEE Trans. Commun.*, vol. 52, pp. 721–727, May 2004.

[56] M. J. Erickson, *Introduction to Combinatorics*. Wiley-Interscience, 1996.

[57] P. J. Cameron, *Combinatorics: Topics, Techniques, Algorithms*. Cambridge University Press, 1996.

[58] J. Garcia-Frias and W. Zhong, "Approaching shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Commun. Lett.*, vol. 7, pp. 266–268, June 2003.

[59] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, pp. 457–458, Mar. 1997.

[60] T. R. Halford, A. J. Grant, and K. M. Chugg, "Which codes have 4-cycle-free tanner graphs?," *IEEE Trans. Inform. Theory*, vol. 52, pp. 4219–4223, Sept. 2006.

[61] R. M. Pyndiah, "Near optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, pp. 1003–1010, Aug. 1998.

[62] D. Rankin, "Single parity check product codes and iterative decoding," *PhD Dissertation, University of Canterbury, Christchurch, New Zealand*, 2001.

[63] D. M. Rankin and T. A. Gulliver, "Single parity check product codes," *IEEE Trans. Commun.*, vol. 49, pp. 1354–1362, Aug. 2001.

[64] A. Ohhashi and T. Ohtsuki, "Performance analysis and code design of low-density parity check (LDPC) coded space-time transmit diversity (STTD) systems," *in Proc. Globecom*, vol. 5, pp. 3118–3122, Nov.-Dec. 2004.

[65] D.-F. Yuan, P. Zhang, and Q. Wang, "The concatenation scheme MLC-STBC combining MLC and STBC over rayleigh fading channels," *in Proc. MILCOM*, vol. 2, pp. 1300–1304, Oct. 2001.

[66] M. Tao and R. S. Cheng, "Generalized layered space-time codes for high data rate wireless communications," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 1067–1075, July 2004.

[67] K. L. Lo, S. Marinkovic, Z. Chen, and B. Vucetic, "Performance comparison of layered space time codes," *Proc. IEEE*, vol. 3, pp. 1382–1387, Apr.-May 2002.

[68] M. Sellathurai and S. Haykin, "A simplified diagonal BLAST architecture with iterative parallel-interference cancellation receivers," *in Proc. ICC*, vol. 10, pp. 3067–3071, June 2001.

[69] S. Marinkovic, B. Vucetic, N. Ishii, S. Yoshida, and A. Ushirokawa, "Space-time iterative and multistage receiver structures for CDMA mobile communication systems," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1594–1604, Aug. 2001.

[70] C. C. Tsimenidis. http://www.staff.ncl.ac.uk/charalampos.tsimenidis/ MatlabFiles/rayleigh.m.

[71] Q. Li, P. Du, and G. Bi, "Generalized soft decision metric generation for MPSK/MQAM without noise variance knowledge," *in Proc. PIMRC*, pp. 1027–1030, Sept. 2003.

[72] L. Hanzo, W. Webb, and T. Keller, *Single- and Multi-Carrier Quadrature Amplitude Modulation*. John Wiley & Sons, Inc., 1 ed., 2000.

[73] R. M. Buehrer and N. A. Kumar, "The impact of channel estimation error on space-time block codes," *in Proc. VTC-Fall*, vol. 3, pp. 1921–1925, Sep. 2002.

[74] D. Mavares and R. P. Torres, "Channel estimation error effects on the performance of STB codes in flat frequency rayleigh channels," *in Proc. VTC-Fall*, vol. 1, pp. 647–651, Oct. 2003.

[75] Z. Diao, D. Shen, and V. O. K. Li, "Performance analysis of space-time codes with channel information errors," *in Proc. VTC-Fall*, vol. 4, pp. 2399–2403, Sep. 2004.

[76] S. J. Grant and J. K. Cavers, "Multiuser channel estimation for detection of cochannel signals," *IEEE Trans. Commun.*, vol. 49, pp. 1845–1855, Oct. 2001.

[77] H. Chen and D. Hong, "Performance analysis of space-time block codes in time-varying rayleigh fading channels," *in Proc. ICASSP*, vol. 3, pp. 2357–2360, May 2002.

[78] F.-C. Zheng and A. G. Burr, "Orthogonal space-time block coding over time-selective fading channels: a PIC detector for the G_i systems," *in Proc. IEEE VTC-Spring*, vol. 1, pp. 389–393, May 2004.

[79] D. Divsalar and M. Simon, "The design of trellis coded MPSK for fading channel: Performance criteria," *IEEE Trans. Commun.*, vol. 36, pp. 1004–1012, Sept. 1988.

[80] P. Sinha, A. Sinha, and D. Basu, "A novel architecture of a re-configurable parallel DSP processor," *in Proc. IEEE-NEWCAS*, pp. 71–74, June 2005.

[81] M. Blaum and R. Roth, "New array codes for multiple phased burst correction," *IEEE Trans. Inform. Theory*, vol. 39, pp. 66–77, Jan. 1993.

[82] J. L. Fan, "Array codes as low-density parity-check codes," *in Proc. 2nd Int. Symp. Turbo Codes and Related Topics, Brest, France*, pp. 553–556, Sep. 2000.

[83] M. P. C. Fossorier and S. Lin, "Computationally efficient soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inform. Theory*, vol. 42, pp. 738–750, May 1996.

[84] O. Milenkovic, N. Kashyap, and D. Leyba, "Shortened array codes of large girth," *IEEE Trans. Inform. Theory*, vol. 52, pp. 3707–3722, Aug. 2006.

[85] T. Okamura, "Designing LDPC codes using cyclic shifts," *in Proc. ISIT*, p. 151, June 2003.

[86] N. Miladinovic and M. Fossorier, "Systematic recursive construction of LDPC codes," *IEEE Communications Letters*, vol. 8, pp. 302–304, May 2004.